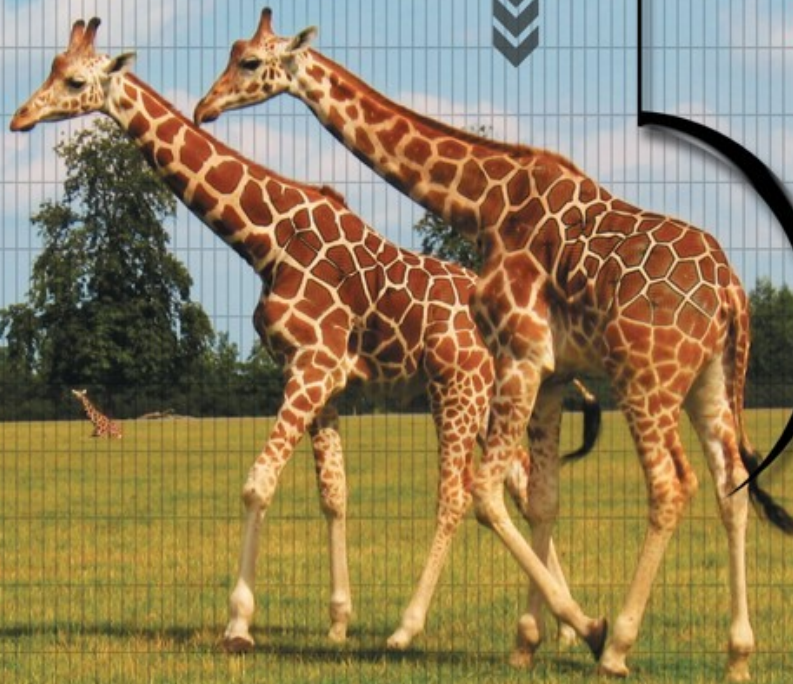


Foundation

# Website Creation

with HTML5, CSS3, and JavaScript

Jonathan Lane  
Tom Barker  
Joe Lewis  
Meitar Moscovitz



**friendsof**   
an Apress® company

*Learn the concepts behind designing  
and developing for the Web*

*Make your mark on the new Web with  
HTML5, CSS3, and JavaScript*

*Design for any device, large or small*

# Foundation Website Creation with HTML5, CSS3, and JavaScript

**Jonathan Lane, Tom Barker, Joseph R. Lewis,  
and Meitar Moscovitz**



# FOUNDATION WEBSITE CREATION WITH HTML5, CSS3, AND JAVASCRIPT

Copyright © 2012 by Jonathan Lane, Tom Barker, Joseph R. Lewis, and Meitar Moscovitz

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 978-1-4302-3789-1

ISBN 978-1-4302-3790-7 (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](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com) or visit [www.apress.com](http://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](http://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](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code](http://www.apress.com/source-code).

## Credits

**President and Publisher:** Paul Manning  
**Copy Editor:** Patrick Meader

**Lead Editor:** Ben Renow-Clarke  
**Compositor:** SPi Global

**Technical Reviewers:** Heather Wallace  
**Proof Reader:** SPi Global

**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,  
**Indexer:** SPi Global

Ben Renow-Clarke, Dominic Shakeshaft,  
**Artist:** SPi Global  
**Cover Image Artist:** Corné van Dooren

Gwenan Spearing, Matt Wade, Tom Welsh  
**Cover Designer:** Anna Ishchenko

**Coordinating Editor:** Debra Kelly

*For my beautiful Rachel.  
You are amazing in so many ways, and I absolutely adore you.  
—Jonathan Lane*

*For my beautiful Rabbit.  
—Tom Barker*

*To my beautiful wife, whom I love beyond measure.  
—Joseph R. Lewis*

*For Aba, whom most of my work on this book is really for.  
Hopefully you'll now feel comfortable using your editor's Source view.  
—Meitar Moscovitz*

# Contents at a Glance

---

About the Authors.....	xii
About the Technical Reviewer.....	xiii
About the Cover Image Designer.....	xiv
Acknowledgments.....	xv
Introduction.....	xvii
<b>Chapter 1: Introducing the Past, Present, and Future of the Web .....</b>	<b>1</b>
<b>Chapter 2: Keeping a Project on Track .....</b>	<b>11</b>
<b>Chapter 3: Planning and High-Level Design .....</b>	<b>31</b>
<b>Chapter 4: Giving Your Pages Structure: HTML5 .....</b>	<b>47</b>
<b>Chapter 5: Exploring Fundamental Concepts of CSS.....</b>	<b>89</b>
<b>Chapter 6: Developing CSS3 in Practice: From Design to Deployment .....</b>	<b>119</b>
<b>Chapter 7: Responsive Design .....</b>	<b>147</b>
<b>Chapter 8: JavaScript Primer.....</b>	<b>159</b>
<b>Chapter 9: A Deeper Dive Into JavaScript .....</b>	<b>175</b>
<b>Chapter 10: Closing the Loop with JavaScript.....</b>	<b>191</b>
<b>Chapter 11: Using Server-Side Technologies .....</b>	<b>219</b>
<b>Chapter 12: Using WordPress to Jumpstart Development.....</b>	<b>235</b>
<b>Chapter 13: Afterword: The Business of the Web .....</b>	<b>253</b>
<b>Index.....</b>	<b>269</b>

# Contents

---

About the Authors.....	xii
About the Technical Reviewer.....	xiii
About the Cover Image Designer.....	xiv
Acknowledgments.....	xv
Introduction.....	xvii
<b>Chapter 1: Introducing the Past, Present, and Future of the Web.....</b>	<b>1</b>
The standard way of doing things .....	2
Every journey starts with a single step: the Web past.....	2
Then there were standards: the Web now .....	5
A crystal ball: the Web future .....	9
What’s inside this book?.....	9
<b>Chapter 2: Keeping a Project on Track .....</b>	<b>11</b>
The traditional approach to project management .....	12
The nine knowledge areas.....	13
Web project management: the power of iteration .....	14
An agile example of planning .....	15
Achieving the goal: identifying doneness .....	16
“But the PMI covers nine areas; you’ve talked about only three!” .....	20
Tools available.....	22
The old toolbox .....	22
The alternatives .....	23
Summary: the best advice .....	26
Profiling professions: Jason Fried.....	26
<b>Chapter 3: Planning and High-Level Design .....</b>	<b>31</b>
The toolbox.....	32
Goals and objectives discussion.....	32
Brainstorming .....	34
User stories and user personas .....	35
Feature/unfeature list.....	37
Wireframes.....	38
Mock-ups .....	39
Information architecture.....	40

Prototype.....	41
<b>Let's go to an example .....</b>	<b>42</b>
<b>Summary: achieving balance .....</b>	<b>43</b>
<b>Profiling professions: Daniel Burka.....</b>	<b>44</b>
<b>Chapter 4: Giving Your Pages Structure: HTML5 .....</b>	<b>47</b>
<b>What are web pages, really?.....</b>	<b>47</b>
<b>The basics of markup.....</b>	<b>48</b>
Elements (or tags).....	48
Attributes and their values.....	51
Empty elements .....	52
<b>Document types .....</b>	<b>52</b>
<b>Starting with HTML5 .....</b>	<b>53</b>
Document shell .....	53
Marking up content.....	55
Links.....	60
The href attribute, URLs, and web page addresses.....	61
Emphasis .....	62
Other inline elements.....	63
Lists.....	64
Images.....	67
I want moving pictures (and sound)!.....	69
Tables.....	70
Forms .....	74
Special characters .....	82
<b>Let's go to an example! .....</b>	<b>83</b>
<b>Chapter 5: Exploring Fundamental Concepts of CSS.....</b>	<b>89</b>
<b>The origins and evolution of Cascading Style Sheets .....</b>	<b>89</b>
CSS1 is born.....	90
Followed quickly by CSS2.....	90
Enter CSS3 .....	91
<b>How CSS works.....</b>	<b>91</b>
Default browser styles .....	91
Anatomy of a style sheet .....	92
The reset style sheet .....	93
<b>Applying styles to web page elements.....</b>	<b>96</b>

---

Inline styles .....	96
Embedded style sheets .....	97
External style sheets .....	97
More CSS selectors: targeting page elements with surgical precision .....	98
Combining multiple CSS selectors in a single rule .....	103
<b>CSS inheritance: making the general case work in your favor.....</b>	<b>103</b>
The CSS cascade and the rules of specificity.....	104
CSS selector specificity.....	105
Source order cascading.....	105
One final way to override.....	107
<b>Visual rendering: document flow and the CSS box model .....</b>	<b>107</b>
What is document flow? .....	108
What is the CSS box model?.....	109
<b>Summary .....</b>	<b>117</b>
<b>Chapter 6: Developing CSS3 in Practice: From Design to Deployment .....</b>	<b>119</b>
<b>The visual source: understanding design documents.....</b>	<b>119</b>
<b>Diving into code: advanced CSS concepts applied .....</b>	<b>120</b>
Page structure: laying out the page .....	124
The header: start at the top .....	126
Exploring the world of fonts.....	127
Room to breathe: space it out .....	128
Give those form elements some style .....	130
Sidebar styling .....	132
CSS3 transitions .....	133
Footer beautification .....	134
All together.....	136
A quick note about browser compatibility .....	141
<b>CSS media types and creating print style sheets.....</b>	<b>141</b>
<b>Designing for other media types and devices .....</b>	<b>146</b>
<b>Summary .....</b>	<b>146</b>
<b>Chapter 7: Responsive Design .....</b>	<b>147</b>
<b>Why bother?.....</b>	<b>147</b>
<b>Strategy and Practice .....</b>	<b>148</b>
<b>Other considerations.....</b>	<b>148</b>
<b>What's next?.....</b>	<b>149</b>



<b>Introducing the @media query</b> .....	<b>149</b>
Adding phone-specific rules .....	152
Adding tablet-specific rules .....	155
Pulling it all together .....	156
<b>Summary</b> .....	<b>158</b>
<b>Chapter 8: JavaScript Primer</b> .....	<b>159</b>
<b>What is JavaScript?</b> .....	<b>159</b>
<b>A brief history of JavaScript</b> .....	<b>160</b>
Early history .....	160
Modern JavaScript .....	161
<b>JavaScript the basics</b> .....	<b>162</b>
Code placement .....	162
Code execution .....	163
Commenting your code .....	163
Expressions, white space, semicolons, and minification .....	164
Variables .....	164
Conditionals and operators .....	167
Iterations .....	171
<b>Summary</b> .....	<b>173</b>
<b>Chapter 9: A Deeper Dive Into JavaScript</b> .....	<b>175</b>
<b>Philosophy of object-oriented design</b> .....	<b>175</b>
Object constructor .....	176
Object notation .....	177
Inheritance .....	178
<b>Functions</b> .....	<b>179</b>
Parameters and arguments .....	180
Return values .....	181
Functions as first class objects .....	182
Anonymous functions .....	183
<b>Interacting with the DOM</b> .....	<b>188</b>
<b>Summary</b> .....	<b>190</b>
<b>Chapter 10: Closing the Loop with JavaScript</b> .....	<b>191</b>
<b>Working with data</b> .....	<b>191</b>
Saving and retrieving client side data .....	192

---

Loading external data .....	203
<b>Tools.....</b>	<b>207</b>
Debugging with firebug.....	207
Minifying JavaScript .....	210
JavaScript Unit Tests with Jasmine .....	212
<b>Summary .....</b>	<b>218</b>
<b>Chapter 11: Using Server-Side Technologies .....</b>	<b>219</b>
<b>The server side removes barriers .....</b>	<b>219</b>
<b>Web servers: dishing out hypertext.....</b>	<b>220</b>
Apache HTTP server.....	220
nginx .....	221
Microsoft Internet Information Services (IIS) .....	221
Mongrel .....	221
A wide range of hosting options.....	221
<b>Databases 101.....</b>	<b>223</b>
Terminology .....	223
The world outside relational databases .....	224
Object databases .....	224
XML databases .....	224
Relational databases .....	225
A look at the RDBMS players .....	227
<b>Other data sources .....</b>	<b>228</b>
<b>Web application languages .....</b>	<b>229</b>
PHP .....	229
Ruby.....	229
Python.....	230
ASP.NET .....	230
Java/JSP .....	230
<b>Frameworks.....</b>	<b>230</b>
Why bother with frameworks?.....	231
A few popular candidates .....	231
CakePHP.....	232
Django.....	232
<b>Summary .....</b>	<b>232</b>
<b>Chapter 12: Using WordPress to Jumpstart Development.....</b>	<b>235</b>

<b>Introducing WordPress .....</b>	<b>236</b>
<b>Installing WordPress .....</b>	<b>236</b>
Creating a database and user .....	237
Grab a copy of WordPress .....	238
A little customization .....	241
<b>Wrapping things up .....</b>	<b>251</b>
<b>Chapter 13: Afterword: The Business of the Web .....</b>	<b>253</b>
<b>Basic needs of the freelance web professional.....</b>	<b>253</b>
Being legally well informed.....	254
Freelancing on the side .....	254
Making the transition .....	254
More information.....	255
<b>Business types .....</b>	<b>255</b>
Sole proprietorship .....	255
Partnership.....	256
Limited Liability Company (in the United States).....	257
Corporation .....	257
<b>Contracts .....</b>	<b>257</b>
Do you need a lawyer? .....	258
Resources? .....	258
<b>Nondisclosure/noncompete.....</b>	<b>259</b>
<b>Making money: financial survival.....</b>	<b>259</b>
Staying in business .....	259
Getting paid (aka accounts receivable).....	260
Tracking time and invoicing .....	260
<b>Do you need an accountant? .....</b>	<b>261</b>
Resources .....	261
<b>Advertising and promotion.....</b>	<b>261</b>
Getting the word out .....	261
<b>Finding work to pay the bills .....</b>	<b>262</b>
Working locally .....	262
Finding work online .....	263
<b>Finding good resources: people .....</b>	<b>264</b>
Hiring: finding the right skills and personality.....	264
Where do you find candidates?.....	264
Finding temporary help: subcontracting .....	265

Partnering with others to complement skill sets ..... 265

**Growing your practice and increasing capacity ..... 266**

**Training to stay current and competitive..... 267**

**Index..... 269**

# About the Authors

---



**Jonathan Lane** works for Harvest (GetHarvest.com) and is the sole employee of Industry Interactive, Inc. He does design and development for a range of clients in Canada and the US, and he produces a few of his own side-projects (like Mailmanagr—[www.mailmanagr.com](http://www.mailmanagr.com)). Jonathan has been working with the Web for more than a decade now in varying capacities.

Jonathan is married and has three kids, Reilly, Parker, and Serenity, which pretty much occupies all of his nonworking time.



**Tom Barker** has been a software engineer since the late '90s, focusing on the full stack of web development. Currently he is the Senior Manager of Web Development at Comcast, an Adjunct Professor at Philadelphia University, a husband, a father, an amateur power lifter, and an arm-chair philosopher. On his team at Comcast are the craftsmen that create and maintain [xfinity.com](http://xfinity.com), and [xfinitytv.com](http://xfinitytv.com). At Philadelphia University, he teaches undergraduate and graduate classes on HTML, JavaScript, ActionScript, PHP, and MySQL. He is obsessed with elegant software

solutions, continual improvement, refining process, data analysis and visualization.



**Joseph R. Lewis** is the Chief Web Architect at Sandia National Laboratories, where his activities include research, development, and instruction of semantic web, social media, and mobile technologies for scientific collaboration and national security applications.

Prior to the computer science career, Joe was a professional classical musician with a masters degree in performance from the New England Conservatory of Music, and he still plays and teaches the double bass when he can.



Born and raised in New York City, **Meitar Moscovitz** first touched a computer when he was an infant in 1986 (an Apple Macintosh Plus). At his father's prompting, he created his first website at the age of 12 and through it created the first online community for teenagers and young adults with bipolar disorder. Out of school by 16, he officially joined the workforce as a junior network administrator; and at 18, he started freelancing full time as a web developer.

After brief excursions into corporate IT with such companies as Apple and Opsware (now HP), he returned to professional web development and worked on websites for clients including Oxygen Media, Inc., and the Institute of Electrical and Electronics Engineers (IEEE).

He now lives in Sydney, Australia, with his brilliant girlfriend of three years, Sara Hames, and works as the senior front-end web developer and IT director for Digital Eskimo, Pty Ltd. In his rapidly diminishing spare time, Meitar enjoys volunteering his technical talents to nonprofit organizations and other small groups. He's also an avid blogger and juggler, and he has way too many profiles on social networking sites.

# About the Technical Reviewer

---

**Heather R. Wallace** is the author of several bestselling titles on a variety of subjects. Her most recent technology related title is *WordPress 3 Site Blueprints* (Packt Publishing, 2010), which shows readers how to build various websites using WordPress along with the right combination of either free or premium themes and/or plugins. Heather has also managed several successful websites and blogs, which she developed using a combination of HTML, CSS, and WordPress.

# About the Cover Image Designer

---



**Corné van Dooren** designed the front cover image for this book. After taking a break from friends of ED to create a new design for the Foundation series, he worked at combining technological and organic forms, with the results now appearing on the cover of this and other books.

Corné spent his childhood drawing on everything at hand and then began exploring the infinite world of multimedia—and his journey of discovery hasn't stopped since. His mantra has always been “the only limit to multimedia is the imagination,” a saying that keeps him moving forward constantly.

Corné works for many international clients, writes features for multimedia magazines, reviews and tests software, authors multimedia studies, and works on many other friends of ED books. If you like Corné's work, be sure to check out his chapter in *New Masters of Photoshop: Volume 2* (friends of ED, 2004). You can see more of his work (and contact him) at his website, [www.cornevandooren.com](http://www.cornevandooren.com).

# Acknowledgments

---

Not a lot of technical books reach a second edition! I was overjoyed to hear that the wonderful folks at Apress wanted to keep the “Foundation Website Creation” train running! Thank you to Ben Renow-Clarke, Matthew Moodie, and Debra Kelly for ushering me through the process for a second round. Thanks to Heather Wallace for all of her excellent and thoughtful feedback in editing this book. Thanks to Tom Barker for seemingly seamlessly filling in all of the gaps in my knowledge and really kicking this edition up a notch. You’re a fantastic writer Tom. Thank you Patrick Meader for dotting our i’s and crossing our t’s. Your feedback was extremely valuable.

Thanks to my co-workers at Harvest, every single one of whom has taught me a lot during my tenure there. I’m honored to get to work with such a fantastic group of people.

Thanks to my wonderful wife, Rachel, for giving me the time and support to write again, and to my kids, Reilly, Parker, and Serenity, for once again keeping the noise level to a dull roar (some of the time). Thanks also to my parents: my mom, Cynthia, for the writing genes, and my father, Rick, for that “get it done” attitude. I’m not sure which was used more on this project.

Jonathan Lane

I want to thank my amazing wife, Lynn, and our beautiful children, Lukas and Paloma, for their patience and understanding while I would write every night until late in the night.

I want to thank Ben Renow-Clarke for thinking of me for this great project. I want to thank Matthew Moodie and Debra Kelly and the rest of the team at Apress for their guidance and direction. And I want to thank Jonathan Lane for being an incredible collaborator.

I want to thank my team at Comcast, every one of you is amazing and I am made better by being a part of such an incredible team.

Tom Barker

To the memory of my teacher, neighbor, mentor, and friend Lester M. Henderson, who taught me that most any problem can be solved with a pencil, a sheet of blank paper, and plenty of determination.

Joseph R. Lewis

First of all, I’d like to acknowledge the web standards giants who came before me, such as Eric Meyer, Molly Holzschlag, Peter-Paul Koch, Jeffrey Zeldman, and others too numerous to mention, without whom I’d have no shoulders to stand on. I hope my contributions to this book have added some real value to the work they produced before me.

I also want to thank all the people who worked with me on this book: my coauthors and technical reviewers, Jonathan Lane, Joe Lewis, and Tom Barker, for their very astute observations and suggestions while I was drafting my contributions; and the staff at Apress/friends of ED.

Thanks also to the wonderful people with whom I work on website projects every day—the staff of Digital Eskimo in Sydney, Australia, for allowing me to take the time off I needed to focus on this work. Even more important, thanks for providing a really great working environment where I can feel like I’m doing the things I want to do instead of the things I need to do to earn a living.



## Acknowledgments

---

I would also like to thank my girlfriend, Sara Hames, the writer. Not only does she deserve thanks for helping proofread all of my work before the copy editor saw any of it and for not being upset with me in spite of getting published before her, but also for her immense support and encouragement in many more important ways than words can ever describe.

Finally, I'd like to thank my family for their support years before I knew I'd ever write a book: my brother, Shir, for being the single most resilient, methodical, and authentic person I know; my mom, Rina, for teaching my brother and I how to be the stalwart people we are (whether we make the choices she'd prefer we make or not); and my father, without whose inspiration, insight, and guidance I may never have found the path that led me to where I want to be today.

Meitar Moscovitz

# Introduction

---

Coming to web development with a blank slate can be pretty intimidating. There are a lot of things to learn about the proper construction of a website, and the most successful websites have a great deal of thought and work put into them before they're even put into production.

Although it can be scary, there has never been a better time to get started than the present. Web browsers are finally starting to reach a point where they all follow web standards (more or less). You have to do less fiddling with things to get them working properly now than ever before. We don't want to jinx it, but we think we can finally start letting our guard down a bit and start trusting browser manufacturers more (yes, even Microsoft).

## Who is this book for?

This book is intended for people who are new to developing for the Web and those who are interested in overhauling their current work to be standards-compliant. It is relevant to individuals working within companies and institutions, as well as for those who freelance.

## How is this book structured?

This book offers a brief history of the World Wide Web and then walks the reader through several chapters on each of the areas relevant to developing a website. Each chapter covers a separate process or technology relevant to working with the Web today.

Readers learn about planning a website, managing the design and development process, and development using web standards; we also provide an overview of server-based technologies and share sample projects along the way.

## Conventions used in this book

To keep this book as clear and easy to follow as possible, the following text conventions are used throughout.

Important words or concepts are normally highlighted on the first appearance in *italic type*.

Code is presented in fixed-width font.

New or changed code is normally presented in **bold fixed-width font**.

Pseudo-code and variable input are written in *italic fixed-width font*.

Menu commands are written in the form Menu ► Submenu ► Submenu.

Where we want to draw your attention to something, we've highlighted it like this:

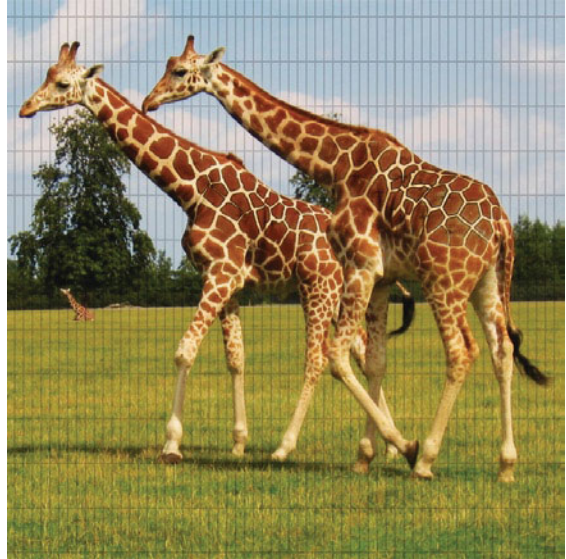
*Ahem, don't say we didn't warn you.*

Sometimes code won't fit on a single line in a book. Where this happens, we use an arrow like this:↪

This is a very, very, very long section of code that should be written all on the same line without ↪ a break.

## Prerequisites

Have you heard of the Web? Ever used Facebook? How about searched for something on Google? Prerequisites passed, this book provides a great introduction to standards-based development for a novice audience. It doesn't stop there, however. Intermediate and advanced readers will also learn new concepts they'll be able to apply to their professional practice.



## Chapter 1

# Introducing the Past, Present, and Future of the Web

---

Believe it or not, when we were kids the standard way to send a message to a friend or family member was by mail. Not e-mail, mind you, but the physical kind requiring a stamp on the envelope. Fax machines came blazing onto the scene and revolutionized communications because, all of a sudden, you could send a document across the country in a matter of minutes, rather than a number of days. Personal computers were starting to show up in houses, but they generally cost an arm and a leg, and they certainly did not have any sort of way of communicating with the outside world. For the most part, assignments in school were handwritten, unless you had a typewriter at home! It was just the standard.

Most people in their twenties today will have a hard time believing that the Internet is a reasonably new invention, and the World Wide Web is even newer. Yet both have had as profound an impact on civilization as the printing press, the steam engine, or the light bulb. When we were growing up, we had an impossible time finding good video games for our PCs. Computers were all about business then. It was easy to find six different word processors, but nearly impossible to find the latest release from Sierra Online (which is owned by Electronic Arts now). These days, if you are looking for a video game, where do you go? The average person will head over to Amazon and preorder their copy of the latest title for next-day shipping. E-commerce has become so ubiquitous that, for certain products, it is preferred over a trip to the local store.

Even between the time that the first edition of this book was written and now, big changes have taken place (and that was just three years ago!). For example, there is now a really good chance that you are not even reading this book in a paper format! You might be using a tablet like an iPad or Playbook, an eReader like Amazon's Kindle, a PDF on your notebook or desktop computer, or even reading this on your cell phone. I have to tell you, it is really an experience to step back and think that in my lifetime, I have gone from a place where 1 in 1,000 households had access to a computer to now, where computer ownership may have peaked and people are moving to simpler, easier devices.

## The standard way of doing things

Because of its highly technical childhood, the World Wide Web has a lot of “hacker baggage.” When we say that, we don't mean that it's a dangerous place where people roam the wires trying to steal your credit card information (although there is some of that, too). We're using the term *hacker* in the classic sense of someone who has a curiosity about technology and tries to find her own way of solving problems. The Web started off very much from a position of trying to solve a problem: distributing research information. It evolved and gained features out of necessity—people needed to be able to add images and tables to their documents. In the early days of mainstream adoption, when businesses started to move online, there wasn't always a perfect way of doing things, so people came up with their own solutions.

A classic example of this is using a table on a web page to lay it out. At one point, we were guilty of this; but at that point we had no choice. Using a table was the only way we could get two columns of text to display on a page side by side. Things are vastly different today, and most would agree that the layout of a page is something better handled in the style information, rather than in the page's markup itself. Web standards have evolved; this isn't 1995 anymore, nor is it even 2005. Things have changed—and, dare we say—improved!

## Every journey starts with a single step: the Web past

It is hard to find a book about Hypertext Markup Language (HTML) and the Web these days that doesn't start off with a history section. We used to wonder why that was the case. To understand why the web standards approach to building websites is the best way to go, you have to know about the progression of the World Wide Web and how it evolved to become what it is today. The World Wide Web has very technical roots, and the trouble (or charm) with most techies is that they like to customize and change things until those things are perfect in their minds. They cannot follow what everyone else is doing; they need to add their own flavor to it.

The Web started its life as a research project at CERN (the European Organization for Nuclear Research) in Geneva, Switzerland. At the time, a researcher there by the name of Tim Berners-Lee was looking for a way to quickly and easily share research findings with his peers and organize information in such a way that it would be easy to archive and retrieve. Although the process was started in the early 1980s, it wasn't until 1990 that Berners-Lee produced the first web server and web browser (which was called WorldWideWeb, the origin of the name).

In 1992, the Web began to hit the mainstream when the National Center for Supercomputing Applications (NCSA) in the United States released Mosaic, which was capable of displaying text and graphics simultaneously. Mosaic gave nonscientists easy access over the Internet to documents created using HTML, what we now call *web pages*. Encouraged by this early success, one of Mosaic's creators, Marc Andreessen, left the NCSA and started Mosaic Communications Corporation. This later became Netscape Communications Corporation, which created Netscape Navigator, the first commercial browser and the web standard-bearer through most of the 1990s.

A company called Spyglass licensed NCSA's source code and eventually released Spyglass Mosaic (although on a newly developed code base). This became the basis for Microsoft Internet Explorer, and it set the stage for the battle for browser supremacy between Netscape and Microsoft.

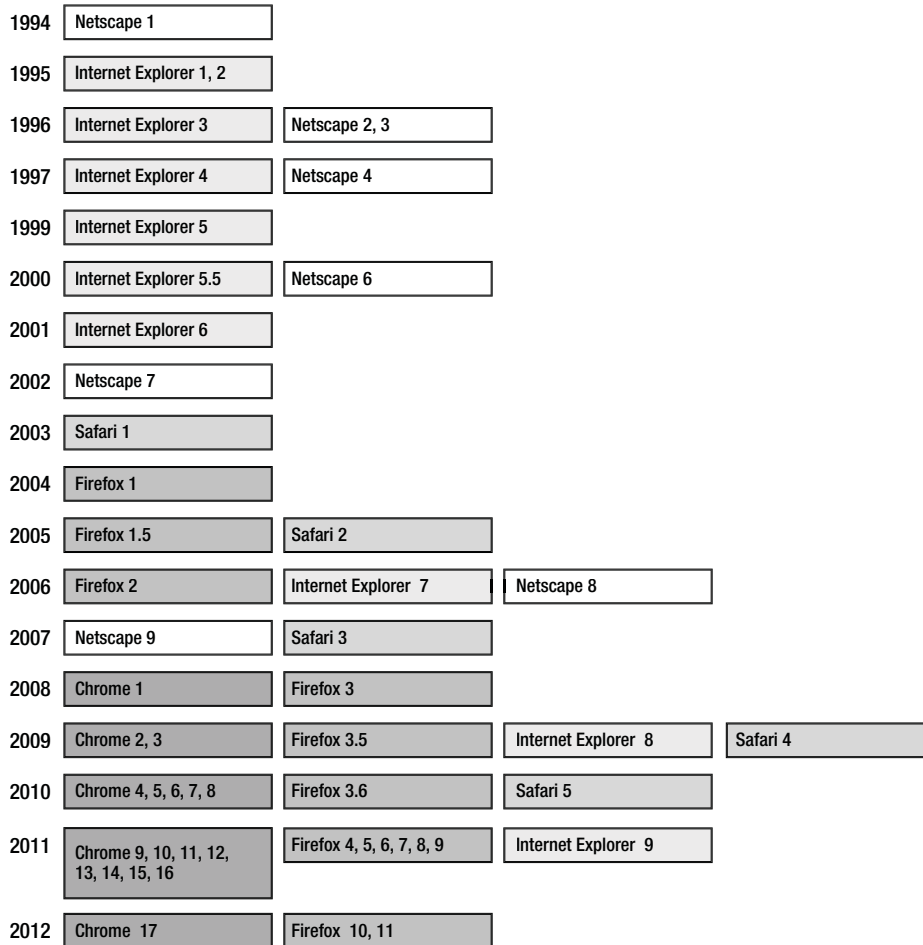
Just prior to the start of the "browser wars" of the 1990s, Berners-Lee recognized that, without some sort of governance, the World Wide Web would experience great challenges as competing web browsers introduced new features. One of the ways to get someone to pick your product over your competition was to offer a bunch of really cool features that the other guy didn't offer. He foresaw compatibility problems emerging as competing companies introduced their own tags in order to add features to the World Wide Web. HTML was the glue binding the Web together, and some central body would need to oversee its evolution in order to maintain a high level of interoperability.

Microsoft gave away Internet Explorer as part of Microsoft Office, bundled it with Windows, and made it available as a free download for all its various operating systems, as well as for Macs. Microsoft was a late starter in the browser market; and by the time it entered the game in 1995, Netscape had an estimated 80 percent market share. The tides rapidly turned, however. By the time Netscape was acquired by America Online in 1998, Microsoft had approximately half of the web browser market. By 2002, Internet Explorer reached its peak with an estimated 96 percent of web surfers using Internet Explorer. In fact, Microsoft so thoroughly thrashed Netscape in the browser wars that the companies ended up in a contracted lawsuit. Eventually, this led to a finding that Microsoft had abused its monopoly power in the marketplace. Figure 1-1 shows a timeline of major browser releases, starting in the early 1990s.

The browser wars were an incredibly difficult time for web developers as manufacturers raced to do their own thing. Versions 3 and 4 of the major browsers often had developers coding two entirely separate versions of their websites in order to ensure compatibility with both Internet Explorer and Netscape. Although web standards existed, most browsers only really supported the basics and relied on competing (and incompatible) technology to do anything advanced, which is what web developers wanted to do.

Netscape, although owned by a fairly major company, started to stagnate. Netscape open sourced its code base, and the Mozilla project was born. The Mozilla community took off with the code and started to really implement standards support in a new way through a complete rewrite of the rendering engine. Although Mozilla was initially based on Netscape Navigator, the tables rapidly turned, and subsequent releases of Netscape became based on Mozilla.

The Mozilla project has forked into a number of new browsers, most notably Firefox, which has sparked a new browser war with Microsoft. Yet this time, the battle has had a positive effect. Microsoft had all but stopped working on Internet Explorer once it released version 6. When Firefox hit the scene with near-perfect support for web standards, it became an immediate hit and gained a great deal of popularity with both



**Figure 1-1.** Timeline of major browser releases

the web cognoscenti and the general public. It seems as though Microsoft had no choice but to improve its web browser, and the result is Internet Explorer 7.

A number of other players emerged in the web browser marketplace, offering competition to the major players. Notably, Opera has always touted excellent support for web standards as one of its major selling features. It has caught on to some degree (2.4 percent as of January 2012), and it is also focusing in large part on mobile devices.

Safari by Apple Computer hit the scene in 2003, and it quickly became one of the most popular choices for Mac users because it integrates well with the operating system and is generally faster than competing browsers on the Mac. Apple introduced a Windows version in 2007. One of the most unique qualities of Safari is that, even with its minor updates (so-called point releases), the browser has improved its support of web standards. This is something that is often relegated to major releases among other browser manufacturers.

Another 800 pound gorilla joined the browser race when Google came on the scene with Chrome in 2008. Chrome took things in a completely different direction and has really shaken up the web browser scene. For the first time since the late nineties, a web browser was created that focused on speed, stability, and security, putting features on the backburner. Sure, Chrome has great standards support and is usually even ahead of the curve when it comes to supporting emerging standards; however, the folks at Google (at least at first) ignored the race to add more to their browser, and instead focused on delivering a blazing fast experience, something that has really resonated with a number of people.

## Then there were standards: the Web now

Imagine if each TV channel was broadcast in either PAL or NTSC in each country, requiring viewers to use a different TV set, depending on what they wanted to watch. Or, imagine writing a research paper, running a search on Google, and then having to open each of the results returned in a different web browser. For example, you would open the first link in Internet Explorer, but the second and third links in Netscape. And for the next link, you would be back to Internet Explorer. Yet, this is pretty much how things worked in the late 1990s. Developers either produced parallel versions of their sites or designed for one specific browser. One of the most common sights on the Web was these wonderful (heavy sarcasm here) graphic badges telling visitors that a given website was “Optimized for Internet Explorer” or “Best Viewed with Netscape Navigator” (see Figure 1-2).



**Figure 1-2.** The “get it now” buttons for Netscape Navigator and Internet Explorer

When we talk about web standards, we’re generally referring to the big three players on the Web: HTML (or XHTML), Cascading Style Sheets (CSS), and JavaScript. The World Wide Web Consortium (W3C) maintains many other standards, such as Scalable Vector Graphics (SVG) for vector graphic display, Portable Network Graphics (PNG) as a way to compress bitmap graphics, and Extensible Markup Language (XML), which is another markup language that is similar in a lot of ways to HTML.

While HTML/XHTML is all about structuring your document, CSS handles enhancing the user experience through formatting and some interactivity. Because they’re two separate pieces of technology, you can physically separate them (by keeping your markup in one file and your style declarations separately in another). The early versions of HTML (prior to the release of CSS) were intended to be everything to everyone. The ability to specify colors and fonts was built right in by specifying the attributes of various tags. As sites began to grow in size, however, the limitations of going this route quickly became apparent.

Imagine that you have put together a website to promote this book. Throughout the site, which covers all the chapters in detail, there are quotes from the book. To really make these quotes jump off the screen, you decide to put them in red and make them a size bigger. In the good old days, your code would look something like this:

```
<font size="+1" color="red">In the good old days,  
► your code would look something like this:</font>
```



With a little HTML and CSS, the code could look like this (note we said “could” because there are multiple ways of doing things—including the old way—but this is a good, structurally relevant example):

```
<blockquote>In the good old days,  
▶ your code would look something like this:</blockquote>
```

And then, in a separate CSS file (or at the top of your HTML document—but again, that’s just another option), you could import/link to pages of your website, which would look something like this:

```
blockquote { font-size: 1.5em; color: red; }
```

So what? They both do the same thing. Web browsers will display the quotes in red and make them nice and big. One of the immediate advantages, however, is in making updates. As we mentioned previously, you are liberally quoting the book throughout the entire website, chapter by chapter. By the end of the project, you discover that you have used more than 300 quotes! The only problem is that we, as the authors (and your client), absolutely hate red text. It makes us so angry. So, we’re going to ask you to change it everywhere.

What would you rather do: update one line in the CSS file or open every page on the site and change the `color` attribute to blue? (Blue is much more calming.) Three hundred quotes—you’re billing by the hour—so that’s not so bad. It’s maybe an extra half day of work. Now extrapolate this example to something like the *New York Times* where there are quite likely millions of quotes throughout the entire website. Are you still keen to go change every font tag on every page?

You might be thinking to yourself, “My software can do search and replace.” This is true. We were once asked to update a department website at an organization where we worked. The website had about 16,000 pages; and by the end of a week, our search and replace still hadn’t finished. We’re not that patient. Are you?

Ease of updating aside, separating content from presentation in this way will help you to maintain consistency across a site. Did you have those quotes set at 1.5em, or 1.2em? It can be hard to remember day after day, week after week, just what sort of formatting parameters you have set up for every bit of text on your web site—so why not remove your memory from the equation? It’s far easier to define a set of style options that can be applied to your document.

Additionally, CSS has evolved and is still evolving, allowing designers and developers to do more and more in terms of the layout and appearance of their pages. Using CSS3, you can position any element anywhere you like on a page and set the opacity of that element (if you have ever used layers in Photoshop, I think you can quickly see the possibilities with this). Although you could probably accomplish the same thing with a table-based layout and transparent graphics, it would probably require you to write a considerable chunk of code in which you would need to set absolute heights and widths of table cells, not to mention all of the time needed to generate and regenerate the graphics as you work through a design.

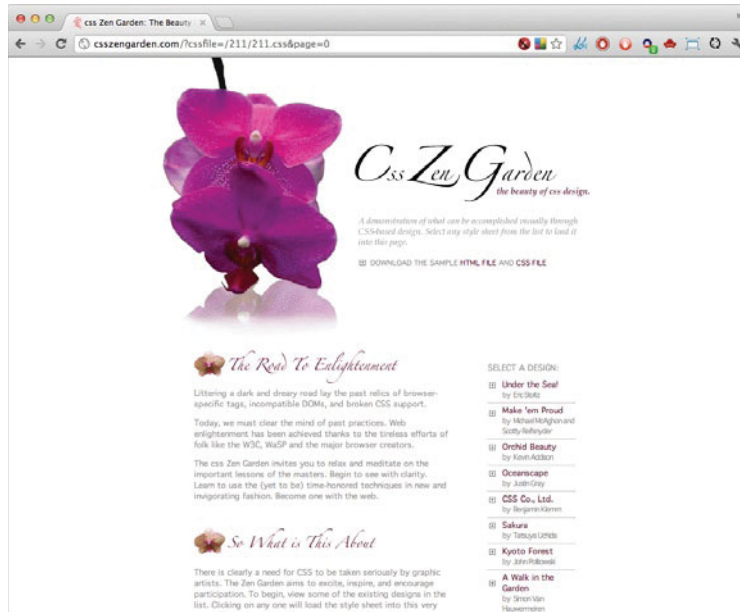
CSS3 has even more formatting options baked in. It lets you specify multiple background images for an element, set the transparency of colors, allow certain elements to be resized on the fly, and even to add drop shadows to text and other elements on the fly. CSS3 takes things one step further by branching out from simple formatting into the “experience” layer. CSS3 lets you perform transforms and transitions on elements; for example, you can rotate and scale them dynamically. Check out the W3C website for more information; or, for a slightly easier to follow format (that includes examples), visit [www.CSS3.info](http://www.CSS3.info) for more information.

The example described previously is a pretty simple representation of what you can do using HTML and CSS. If you head over to the CSS Zen Garden (<http://CSSzengarden.com/>) and click through some of the examples there, you can gain an appreciation for the level of control you can achieve by tweaking the CSS for a website. CSS Zen Garden uses the same markup for every example (the XHTML part stays the same on every page). The only thing that is different on each page is the CSS file that is included, as well as any supporting graphics (see Figure 1-3, Figure 1-4, and Figure 1-5).



**Figure 1-3.** An example from CSS Zen Garden (Make 'em Proud), which you can find at [www.CSSzengarden.com](http://www.CSSzengarden.com). It uses the same XHTML markup as Figures 1-4 and 1-5, but each page is styled using different CSS style sheets and images.

Imagine not having to touch a single page of markup when redesigning a website and still getting a completely different end result. Properly structured, standards-based pages will let you do that. This could be a major advantage for companies building large-scale content management systems: simply change the style sheet, and the CMS has been tailored to a new customer. Some companies are doing this, but others haven't caught on to this, yet.



**Figure 1-4.** This is the same page as shown in Figure 1-3, but styled using an alternate style sheet (Orchid Beauty) at the CSS Zen Garden.



**Figure 1-5.** A final example (CSS Co., Ltd.) that shows just how powerful CSS can be for changing the look of a page.

## A crystal ball: the Web future

The Web is a cool place in many ways, not least because we don't need to be a fortune-teller to see into the future. Because standards take time to develop, we already have an idea of what the future holds. HTML5 is the new hotness, but it is still an evolving standard in the world of markup. The majority of the specification has now been defined with the goal of tying up a lot of the loose ends from HTML 4 and bridging the gap with the world of XHTML.

The W3C process for standards approval goes through numerous steps, and it can easily span several years. The final product of this process is a W3C Recommendation. Because the W3C isn't a governing body, all it can do is make a recommendation that software companies can then choose to follow. There are no "laws" of the Web; so if Apple were to decide it wants to drop HTML support in Safari in favor of its own markup language, it could. As you can see from our previous history lesson, this wouldn't have been completely out of the question in the 1990s; but today, it would probably be very self-defeating. The culture of the Web is all about interoperability; and when companies attempt to limit interoperability to gain some sort of competitive advantage, we all lose.

This has led to somewhat mixed results. On one hand, you can go to almost any website and have it work in almost any web browser. There are still some companies that are pulling the old "Please download Internet Explorer" trick, but those companies are few and far between (my bank just pulled this the other day on me, and I am now looking for a new bank). The fact of the matter is that interoperability, for businesses, means a larger potential customer base. For most Mac users, if they visit a website that is "Internet Explorer-only," they're dead in the water. This is true for mobile users, the fastest-growing segment of Internet users, as well. Sure, they could all buy a Windows machine or run some virtualization software to make it work; but chances are, if they have the choice (and the Web is all about choices), they just won't bother. They'll go find a competitor that supports Chrome, Safari, or Firefox.

The consultation process for developing new standards and revising old ones is very thorough, and it involves a lot of consultation with the community. Several drafts are developed, a great deal of feedback is collected, and eventually a Recommendation is put forth. We've presented HTML and XHTML almost interchangeably in this first chapter. And while both HTML and XHTML are valid web standards, in this book we'll focus on HTML5, the latest in a long line of markup language recommendations.

HTML5 was originally targeted at web applications with an eye to improving speed, compatibility, and the ease with which you can construct them. It introduces a number of new elements and relaxes some of the strict rules around markup in XHTML, making it easier to learn than its predecessors. HTML5 also attempts to minimize the need for a lot of third-party plugins for displaying content.

## What's inside this book?

Our goal is to take this book beyond web standards to cover the entire process of developing a website based on standards. Believe it or not, there is more to putting together a great website than just knowing a little HTML and CSS. This introduction will expose you to different techniques and technologies, and we hope it will encourage you to pursue those areas you find most compelling.

The publishing industry as a whole is changing and evolving. Websites, and other electronic publishing media are becoming more and more prominent as they drive costs down, facilitate and speed up the process of creating content, and allow instant delivery of that content. This is having a profound effect in a

number of related industries as advertising dollars are shifted out of newspapers and magazines toward online venues.

Technologies come and go or—as in the case of HTML—evolve. Three years ago, we wrote that “Two years from now, we may have to completely rewrite sections of this book.” That has proven itself true! While everything in the original edition of this book will still work, the techniques presented there are not the fastest, easiest, or best ways to do things anymore. It’s exciting to be working with ever-changing Internet and web technologies. When the original edition was written, the mobile web was just on the horizon and hadn’t fully developed. There’s no telling what we’ll find around the next corner.