HTML5 & CSS3 FOR THE REAL WORLD

BY ALEXIS GOLDSTEIN
LOUIS LAZARIS
ESTELLE WEYL

POWERFUL HTML5 AND CSS3 TECHNIQUES YOU CAN USE TODAY!

Summary of Contents

sitepoint®

HTML5 & CSS3 FOR THE REAL WORLD

BY **ALEXIS GOLDSTEIN**
LOUIS LAZARIS
ESTELLE WEYL

HTML5 & CSS3 for the Real World

by Alexis Goldstein, Louis Lazaris, and Estelle Weyl

Program Director: Lisa Lang
Technical Editor: Louis Simoneau
Expert Reviewer: Russ Weakley
Indexer: Michele Combs
Editor: Kelly Steele
Cover Design: Alex Walker
Printing History:
First Edition: May 2011

Notice of Rights

Notice of Liability

The author and publisher have made every effort to ensure the accuracy of the information herein. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors and SitePoint Pty. Ltd., nor its dealers or distributors, will be held liable for any damages caused either directly or indirectly by the instructions contained in this book, or by the software or hardware products described herein.

Trademark Notice

Rather than indicating every occurrence of a trademarked name as such, this book uses the names only in an editorial fashion and to the benefit of the trademark owner with no intention of infringement of the trademark.

Published by SitePoint Pty. Ltd.

48 Cambridge Street, Collingwood
VIC 3066 Australia
Web: www.sitepoint.com
Email: business@sitepoint.com

ISBN 978-0-9808469-0-4
Printed and bound in the United States of America

About Alexis Goldstein

Alexis Goldstein first taught herself HTML while a high school student in the mid-1990s, and went on to get her degree in Computer Science from Columbia University. She runs her own software development and training company, aut faciam LLC. Before striking out on her own, Alexis spent seven years in technology on Wall Street, where she worked in both the cash equity and equity derivative spaces at three major firms, and learned to love daily code reviews. She is a teacher and co-organizer of Girl Develop It, a group that conducts low-cost programming classes for women, and a very proud member of the NYC Resistor hacker-space in Brooklyn, NY. You can find Alexis at her website, http://alexisgo.com/.

About Louis Lazaris

Louis Lazaris is a freelance web designer and front-end developer based in Toronto, Canada who has been involved in the web design industry since 2000. Louis has been working on websites ever since the days when table layouts and one-pixel GIFs dominated the industry. Over the past five years he has transitioned to embrace web standards while endeavoring to promote best practices that help both developers and their clients reach practical goals for their projects. Louis writes regularly for a number of top web design blogs including his own site, Impressive Webs (http://www.impressivewebs.com/.

About Estelle Weyl

Estelle Weyl is a front-end engineer from San Francisco who has been developing standards-based accessible websites since 1999. Estelle began playing with CSS3 when the iPhone was released in 2007, and after four years of web application development for mobile WebKit, she knows (almost) every CSS3 quirk on WebKit, and has vast experience implementing components of HTML5. She writes two popular technical blogs with tutorials and detailed grids of CSS3 and HTML5 browser support (http://www.standardista.com/). Estelle's passion is teaching web development, where you'll find her speaking on CSS3, HTML5, JavaScript, and mobile web development at conferences around the USA (and, she hopes, the world).

About the Expert Reviewer

Russ Weakley has worked in the design field for over 18 years, primarily in web design and development, and web training. Russ co-chairs the Web Standards Group and is a founding committee member of the Web Industry Professionals Association of Australia (WIPA). Russ has produced a series of widely acclaimed CSS tutorials, and is internationally recognized for his presentations and workshops. He manages Max Design (http://maxdesign.com.au/).

About SitePoint

SitePoint specializes in publishing fun, practical, and easy-to-understand content for web professionals. Visit http://www.sitepoint.com/ to access our blogs, books, newsletters, articles, and community forums.

To my parents, who always encourage and believe in me.

And to my talented, prolific, and loving Grandma Joan. You always keep me painting, no matter what else I may be doing.

—Alexis

To Melanie, the best cook in the world.

And to my parents, for funding the original course that got me into this unique industry.

—Louis

To Amie, for putting up with me, and to Spazzo and Puppers, for snuggling with me as I worked away.

—Estelle

Table of Contents

Foreword

Heard of Sjoerd Visscher? I'd venture to guess you haven't, but what he considered a minor discovery is at the foundation of our ability to use HTML5 today.

Back in 2002, in The Hague, Netherlands, Mr. Visscher was attempting to improve the performance of his XSL output. He switched from `createElement` calls to setting the `innerHTML` property, and then realized that all the unknown, non-HTML elements were no longer able to be styled by CSS.

Fast forward to 2008, and HTML5 is gaining momentum. New elements have been specified, but in practice Internet Explorer versions 6-8 pose a problem, as they fail to recognize unknown elements; the new elements are unable to hold children and CSS has no effect on them. This depressing fact was posing quite a hindrance to HTML5 adoption.

Now, half a decade after his discovery, Sjoerd innocently mentions this trick in a comment on the blog of the W3C HTML Working Group co-chair, Sam Ruby: "BTW, if you want CSS rules to apply to unknown elements in IE, you just have to do `document.createElement(elementName)`. This somehow lets the CSS engine know that elements with that name exist."

Ian Hickson, lead editor of the HTML5 spec, was as surprised as the rest of the Web. Having never heard of this trick before, he was happy to report: "This piece of information makes building an HTML5 compatibility shim for IE7 far easier than had previously been assumed."

A day later, John Resig wrote the post that coined the term "HTML5 shiv." Here's a quick timeline of what followed:

- January 2009: Remy Sharp creates the first distributable script for enabling HTML5 element use in IE.
- June 2009: Faruk Ateş includes the HTML5 shiv in Modernizr's initial release.
- February 2010: A ragtag team of superstar JavaScript developers including Remy, Kangax, John-David Dalton, and PorneL collaborate and drop the file size of the script.

- March 2010: Mathias Bynens and others notice that the shiv doesn't affect pages printed from IE. It was a sad day. I issue an informal challenge to developers to find a solution.
- April 2010: Jonathan Neal answers that challenge with the IE Print Protector (IEPP), which captured the scope of the HTML5 shiv but added in support for printing the elements as well.
- April 2010: Remy replaces the legacy HTML5 shiv solution with the new IEPP.
- February 2011: Alexander Farkas carries the torch, moving the IEPP project to GitHub, adding a test suite, fixing bugs, and improving performance.
- April 2011: IEPP v2 comes out. Modernizr and the HTML5 shiv inherit the latest code while developers everywhere continue to use HTML5 elements in a cross-browser fashion without worry.

The tale of the HTML5 shiv is just one example of community contribution that helps to progress the open web movement. It's not just the W3C or the browsers who directly affect how we work on the Web, but people like you and me. I hope this book encourages you to contribute in a similar manner; the best way to further your craft is to actively share what you learn.

Adopting HTML5 and CSS3 today is easier than ever, and seriously fun. This book presents a wealth of practical information that gives you what you need to know to take advantage of HTML5 now. The authors—Alexis, Louis, and Estelle—are well-respected web developers who present a realistic learning curve for you to absorb the best practices of HTML5 development easily.

I trust this book is able to serve you well, and that you'll be as excited about the next generation of the Web as I am.

—

Paul Irish
jQuery Dev Relations,
Lead Developer of Modernizr and HTML5 Boilerplate
April 2011

Preface

Welcome to *HTML5 & CSS3 for the Real World*. We're glad you've decided to join us on this journey of discovering some of the latest and the greatest in front-end website building technology.

If you've picked up a copy of this book, it's likely that you've dabbled to some degree in HTML and CSS. You might even be a bit of a seasoned pro in certain areas of markup, styling, or scripting, and now want to extend those skills further by dipping into the new features and technologies associated with HTML5 and CSS3.

Learning a new task can be difficult. You may have limited time to invest in poring over the official documentation and specifications for these web-based languages. You also might be turned off by some of the overly technical books that work well as references but provide little in the way of real-world, practical examples.

To that end, our goal with this book is to help you learn through hands-on, practical instruction that will assist you to tackle the real-world problems you face in building websites today—with a specific focus on HTML5 and CSS3.

But this is more than just a step-by-step tutorial. Along the way, we'll provide plenty of theory and technical information to help fill in any gaps in your understanding—the whys and hows of these new technologies—while doing our best not to overwhelm you with the sheer volume of cool new stuff. So let's get started!

Who Should Read This Book

This book is aimed at web designers and front-end developers who want to learn about the latest generation of browser-based technologies. You should already have at least intermediate knowledge of HTML and CSS, as we won't be spending any time covering the basics of markup and styles. Instead, we'll focus on teaching you what new powers are available to you in the form of HTML5 and CSS3.

The final two chapters of this book cover some of the new JavaScript APIs that have come to be associated with HTML5. These chapters, of course, require some basic familiarity with JavaScript—but they're not critical to the rest of the book. If you're unfamiliar with JavaScript, there's no harm in skipping over them for now, returning later when you're better acquainted with it.

What's in This Book

This book comprises eleven chapters and three appendices. Most chapters follow on from each other, so you'll probably get the most benefit reading them in sequence, but you can certainly skip around if you only need a refresher on a particular topic.

Chapter 1: *Introducing HTML5 and CSS3*

Before we tackle the hands-on stuff, we'll present you with a little bit of history, along with some compelling reasons to start using HTML5 and CSS3 today. We'll also look at the current state of affairs in terms of browser support, and argue that a great deal of these new technologies are ready to be used today—so long as they're used wisely.

Chapter 2: *Markup, HTML5 Style*

In this chapter, we'll show you some of the new structural and semantic elements that are new in HTML5. We'll also be introducing *The HTML5 Herald*, a demo site we'll be working on throughout the rest of the book. Think `div`s are boring? So do we. Good thing HTML5 now provides an assortment of options: `article`, `section`, `nav`, `footer`, `aside`, and `header`!

Chapter 3: *More HTML5 Semantics*

Continuing on from the previous chapter, we turn our attention to the new way in which HTML5 constructs document outlines. Then we look at a plethora of other semantic elements that let you be a little more expressive with your markup.

Chapter 4: *HTML5 Forms*

Some of the most useful and currently applicable features in HTML5 pertain to forms. A number of browsers now support native validation on email types like emails and URLs, and some browsers even support native date pickers, sliders, and spinner boxes. It's almost enough to make you enjoy coding forms! This chapter covers everything you need to know to be up to speed writing HTML5 forms, and provides scripted fallbacks for older browsers.

Chapter 5: *HTML5 Audio and Video*

HTML5 is often touted as a contender for the online multimedia content crown, long held by Flash. The new `audio` and `video` elements are the reason—they provide native, scriptable containers for your media without relying on a third-party plugin like Flash. In this chapter, you'll learn all the ins and outs of putting these new elements to work.

Chapter 6: *Introducing CSS3*

Now that we've covered just about all of HTML5, it's time to move onto its close relative CSS3. We'll start our tour of CSS3 by looking at some of the new selectors that let you target elements on the page with unprecedented flexibility. Then we'll follow up with a look at some new ways of specifying color in CSS3, including transparency. We'll close the chapter with a few quick wins—cool CSS3 features that can be added to your site with a minimum of work: text shadows, drop shadows, and rounded corners.

Chapter 7: *CSS3 Gradients and Multiple Backgrounds*

When was the last time you worked on a site that didn't have a gradient or a background image on it? CSS3 provides some overdue support to developers spending far too much time wrangling with Photoshop, trying to create the perfect background gradients and images without breaking the bandwidth bank. Now you can specify linear or radial gradients right in your CSS without images, and you can give an element any number of background images. Time to ditch all those spare `div`s you've been lugging around.

Chapter 8: *CSS3 Transforms and Transitions*

Animation has long been seen as the purview of JavaScript, but CSS3 lets you offload some of the heavy lifting to the browser. Transforms let you rotate, flip, skew, and otherwise throw your elements around. Transitions can add some subtlety to the otherwise jarring all-on or all-off state changes we see on our sites. We wrap up this chapter with a glimpse of the future; while CSS keyframe animations still lack widespread support, we think you'll agree they're pretty sweet.

Chapter 9: *Embedded Fonts and Multicolumn Layouts*

Do you prefer Arial or Verdana? Georgia or Times? How about none of those? In this chapter, we'll look at how we can move past the "web-safe" fonts of yesteryear and embed any fonts right into our pages for visitors to download along with our stylesheets and images. We'll also look at a promising new CSS feature that allows us to lay out content across multiple columns without using extra markup or the dreaded `float`.

Chapter 10: *Geolocation, Offline Web Apps, and Web Storage*

The latest generation of browsers come equipped with a wide selection of new standard JavaScript APIs. Many of these are specifically geared towards mobile browsers, but still carry benefits for desktop users. In this chapter, we'll look at three of the most exciting: Geolocation, Offline Web Apps, and Web Storage. We'll also touch briefly on some of the APIs that we won't be covering in detail—either because they're poorly supported, or have limited use cases—and give you some pointers should you want to investigate further.

Chapter 11: *Canvas, SVG, and Drag and Drop*

We devote the book's final chapter to, first of all, covering two somewhat competing technologies for drawing and displaying graphics. Canvas is new to HTML5, and provides a pixel surface and a JavaScript API for drawing shapes to it. SVG, on the other hand, has been around for years, but is now achieving very good levels of browser support, so it's an increasingly viable alternative. Finally, we'll cover one more new JavaScript API—Drag and Drop—which provides native handling of drag-and-drop interfaces.

Appendix A: *Modernizr*

A key tool in any HTML5 superhero's utility belt, Modernizr is a nifty little JavaScript library that detects support for just about every HTML5 and CSS3 feature, allowing you to selectively style your site or apply fallback strategies. We've included a quick primer on how to use Modernizr in this appendix, even though Modernizr is used throughout the book. This way, you have a ready reference available in one place, while the other chapters focus on the meat of HTML5 and CSS3.

Appendix B: *WAI-ARIA*

A separate specification that's often mentioned in the same breath as HTML5, WAI-ARIA is the latest set of tools to help make sophisticated web applications accessible to users of assistive technology. While a whole book could be devoted to WAI-ARIA, we thought it beneficial to include a quick summary of what it is, as well as some pointers to places where you can learn more.

Appendix C: *Microdata*

Microdata is part of the HTML5 specification that deals with annotating markup with machine-readable labels. It's still somewhat in flux, but we thought it was worthwhile to get you up to speed with a few examples.

Where to Find Help

SitePoint has a thriving community of web designers and developers ready and waiting to help you out if you run into trouble. We also maintain a list of known errata for the book, which you can consult for the latest updates.

The SitePoint Forums

The SitePoint Forums[1] are discussion forums where you can ask questions about anything related to web development. You may, of course, answer questions too. That's how a forum site works—some people ask, some people answer, and most people do a bit of both. Sharing your knowledge benefits others and strengthens the community. A lot of interesting and experienced web designers and developers hang out there. It's a good way to learn new stuff, have questions answered in a hurry, and generally have a blast.

The Book's Website

Located at http://sitepoint.com/books/rw1/, the website that supports this book will give you access to the following facilities:

The Code Archive

As you progress through this book, you'll note a number of references to the code archive. This is a downloadable ZIP archive that contains every line of example source code printed in this book. If you want to cheat (or save yourself from carpal tunnel syndrome), go ahead and download the archive.[2]

Updates and Errata

No book is perfect, and we expect that watchful readers will be able to spot at least one or two mistakes before the end of this one. The Errata page[3] on the book's

[1] http://www.sitepoint.com/forums/

[2] http://www.sitepoint.com/books/rw1/code.php

[3] http://www.sitepoint.com/books/rw1/errata.php

website will always have the latest information about known typographical and code errors.

The SitePoint Newsletters

In addition to books like this one, SitePoint publishes free email newsletters, such as the *SitePoint Tech Times*, *SitePoint Tribune*, and *SitePoint Design View*, to name a few. In them, you'll read about the latest news, product releases, trends, tips, and techniques for all aspects of web development. Sign up to one or more SitePoint newsletters at http://www.sitepoint.com/newsletter/.

The SitePoint Podcast

Join the SitePoint Podcast team for news, interviews, opinion, and fresh thinking for web developers and designers. We discuss the latest web industry topics, present guest speakers, and interview some of the best minds in the industry. You can catch up on the latest and previous podcasts at http://www.sitepoint.com/podcast/, or subscribe via iTunes.

Your Feedback

If you're unable to find an answer through the forums, or if you wish to contact us for any other reason, the best place to write is `books@sitepoint.com`. We have a well-staffed email support system set up to track your inquiries, and if our support team members can't answer your question, they'll send it straight to us. Suggestions for improvements, as well as notices of any mistakes you may find, are especially welcome.

Acknowledgments

Alexis Goldstein

Thank you to Lisa Lang, Russ Weakley, and Louis Simoneau. Your attention to detail, responsiveness, and impressive technical expertise made this book an absolute joy to work on. Thank you to my co-authors, Louis and Estelle, who never failed to impress me with their deep knowledge, vast experience, and uncanny ability to find bugs in the latest browsers. A special thank you to Estelle for the encouragement, for which I am deeply grateful. Finally, thank you to my girlfriend Tabatha, who now knows more about HTML5's JavaScript APIs than most of my nerdy friends. Thank you for your patience, your feedback, and all your support. You help me take things less seriously, which, as anyone who knows me knows, is a monumental task. Thank you for always making me laugh.

Louis Lazaris

Thank you to my wife for putting up with my odd work hours while I took part in this great project. Thanks to my talented co-authors, Estelle and Alexis, for gracing me with the privilege of having my name alongside theirs, and, of course, to our expert reviewer Russ for his great technical insight during the writing process. And special thanks to the talented staff at SitePoint for their super-professional handling of this project and everything that goes along with such an endeavor.

Estelle Weyl

Thank you to the entire open source community. With the option to "view source," I have learned from every developer who opted for markup rather than plugins. I would especially like to thank Jen Mei Wu and Sandi Watkins, who helped point me in the right direction when I began my career. Thank you to Dave Gregory and Laurie Voss who have always been there to help me find the words when they escaped me. Thank you to Stephanie Sullivan for brainstorming over code into the wee hours of the morning. And thank you to my developer friends at Opera, Mozilla, and Google for creating awesome browsers, providing us with the opportunity to not just play with HTML5 and CSS, but also to write this book.

Conventions Used in This Book

You'll notice that we've used certain typographic and layout styles throughout the book to signify different types of information. Look out for the following items:

Code Samples

Code in this book will be displayed using a fixed-width font, like so:

```
<h1>A Perfect Summer's Day</h1>
<p>It was a lovely day for a walk in the park. The birds
were singing and the kids were all back at school.</p>
```

If the code is to be found in the book's code archive, the name of the file will appear at the top of the program listing, like this:

example.css

```
.footer {
  background-color: #CCC;
  border-top: 1px solid #333;
}
```

If only part of the file is displayed, this is indicated by the word *excerpt*:

example.css ***(excerpt)***

```
  border-top: 1px solid #333;
```

If additional code is to be inserted into an existing example, the new code will be displayed in bold:

```
function animate() {
  new_variable = "Hello";
}
```

Where existing code is required for context, rather than repeat all the code, a vertical ellipsis will be displayed:

```
function animate() {
  ⋮
  return new_variable;
}
```

Some lines of code are intended to be entered on one line, but we've had to wrap them because of page constraints. A ➥ indicates a line break that exists for formatting purposes only, and should be ignored:

```
URL.open("http://www.sitepoint.com/blogs/2007/05/28/user-style-she
➥ets-come-of-age/");
```

Tips, Notes, and Warnings

Hey, You!

Tips will give you helpful little pointers.

Ahem, Excuse Me …

Notes are useful asides that are related—but not critical—to the topic at hand. Think of them as extra tidbits of information.

Make Sure You Always …

… pay attention to these important points.

Watch Out!

Warnings will highlight any gotchas that are likely to trip you up along the way.

Chapter 1

Introducing HTML5 and CSS3

This chapter gives a basic overview of how we arrived where we are today, why HTML5 and CSS3 are so important to modern websites and web apps, and how using these technologies will be invaluable to your future as a web professional.

Of course, if you'd prefer to just get into the meat of the project that we'll be building, and start learning how to use all the new bells and whistles that HTML5 and CSS3 bring to the table, you can always skip ahead to Chapter 2 and come back later.

What is HTML5?

What we understand today as HTML5 has had a relatively turbulent history. You probably already know that HTML is the predominant markup language used to describe content, or data, on the World Wide Web. HTML5 is the latest iteration of that markup language, and includes new features, improvements to existing features, and scripting-based APIs.

That said, HTML5 is not a reformulation of previous versions of the language—it includes all valid elements from both HTML4 and XHTML 1.0. Furthermore, it's been designed with some primary principles in mind to ensure it works on just about every platform, is compatible with older browsers, and handles errors gracefully. A summary of the design principles that guided the creation of HTML5 can be found on the W3C's HTML Design Principles page[1].

First and foremost, HTML5 includes redefinitions of existing markup elements, and new elements that allow web designers to be more expressive in the semantics of their markup. Why litter your page with `div`s when you can have `article`s, `section`s, `header`s, `footer`s, and more?

The term "HTML5" has additionally been used to refer to a number of other new technologies and APIs. Some of these include drawing with the `<canvas>` element, offline storage, the new `<video>` and `<audio>` elements, drag-and-drop functionality, Microdata, embedded fonts, and others. In this book, we'll be covering a number of those technologies, and more.

What's an API?

API stands for Application Programming Interface. Think of an API the same way you think of a graphical user interface—except that instead of being an interface for humans, it's an interface for your code. An API provides your code with a set of "buttons" (predefined methods) that it can press to elicit the desired behavior from the system, software library, or browser.

API-based commands are a way of abstracting the more complex stuff that's done in the background (or sometimes by third-party software). Some of the HTML5-related APIs will be introduced and discussed in later sections of this book.

Overall, you shouldn't be intimidated if you've had little experience with JavaScript or any scripting-related APIs. While it would certainly be beneficial to have some experience with JavaScript, it isn't mandatory.

Whatever the case, we'll walk you through the scripting parts of our book gradually, to ensure you're not left scratching your head!

It should also be noted that some of the technologies that were once part of HTML5 have been separated from the specification, so technically, they no longer fall under the "HTML5" umbrella. Certain other technologies were *never* part of HTML5, yet have at times been lumped in under the same label. This has instigated the use of

[1] http://www.w3.org/TR/html-design-principles/

broad, all-encompassing expressions such as “HTML5 and related technologies.” Bruce Lawson even half-jokingly proposed the term “NEWT” (New Exciting Web Technologies)[2] as an alternative.

However, in the interest of brevity—and also at the risk of inciting heated arguments—we’ll generally refer to these technologies collectively as “HTML5.”

How did we get here?

The web design industry has evolved in a relatively short time period. Twelve years ago, a website that included images and an eye-catching design was considered “top of the line” in terms of web content.

Now, the landscape is quite different. Simple, performance-driven, Ajax-based web apps that rely on client-side scripting for critical functionality are becoming more and more common. Websites today often resemble standalone software applications, and an increasing number of developers are viewing them as such.

Along the way, web markup evolved. HTML4 eventually gave way to XHTML, which is really just HTML 4 with strict XML-style syntax. Currently, both HTML 4 and XHTML are in general use, but HTML5 is gaining headway.

HTML5 originally began as two different specifications: Web Forms 2.0 and Web Apps 1.0. Both were a result of the changed web landscape, and the need for faster, more efficient, maintainable web applications. Forms and app-like functionality are at the heart of web apps, so this was the natural direction for the HTML5 spec to take. Eventually, the two specs were merged to form what we now call HTML5.

During the time that HTML5 was in development, so was XHTML 2.0. That project has since been abandoned to allow focus on HTML5.

Would the real HTML5 spec please stand up?

Because the HTML5 specification is being developed by two different bodies (the WHATWG and the W3C), there are two different versions of the spec. The W3C (or World Wide Web Consortium) you’re probably familiar with: it’s the organization that maintains the original HTML and CSS specifications, as well as a host of other

[2] http://www.brucelawson.co.uk/2010/meet-newt-new-exciting-web-technologies/

web-related standards, such as SVG (scalable vector graphics) and WCAG (web content accessibility guidelines.)

The WHATWG (aka the Web Hypertext Application Technology Working Group), on the other hand, might be new to you. It was formed by a group of people from Apple, Mozilla, and Opera after a 2004 W3C meeting left them disheartened. They felt that the W3C was ignoring the needs of browser makers and users by focusing on XHTML 2.0, instead of working on a backwards-compatible HTML standard. So they went off on their own and developed the Web Apps and Web Forms specifications discussed above, which were then merged into a spec they called HTML5. On seeing this, the W3C eventually gave in and created its own HTML5 specification based on the WHATWG's spec.

This can seem a little confusing. Yes, there are some politics behind the scenes that we, as designers and developers, have no control over. But should it worry us that there are two versions of the spec? In short, no.

The WHATWG's version of the specification can be found at http://www.whatwg.org/html/, and has recently been renamed "HTML" (dropping the "5"). It's now called a "living standard," meaning that it will be in constant development and will no longer be referred to using incrementing version numbers.[3]

The WHATWG version contains information covering HTML-only features, including what's new in HTML5. Additionally, there are separate specifications being developed by the WHATWG that cover the related technologies. These specifications include Microdata, Canvas 2D Context, Web Workers, Web Storage, and others.[4]

The W3C's version of the spec can be found at http://dev.w3.org/html5/spec/, and the separate specifications for the other technologies can be accessed through http://dev.w3.org/html5/.

So what's the difference between the W3C spec and that of WHATWG? Briefly, the WHATWG version is a little more informal and experimental (and, some might argue, more forward-thinking). But overall, they're very similar, so either one can be used as a basis for studying new HTML5 elements and related technologies.

[3] See http://blog.whatwg.org/html-is-the-new-html5/ for an explanation of this change.

[4] For details, see http://wiki.whatwg.org/wiki/FAQ#What_are_the_various_versions_of_the_spec.3F.

Why should I care about HTML5?

As mentioned, at the core of HTML5 are a number of new semantic elements, as well as several related technologies and APIs. These additions and changes to the language have been introduced with the goal of web pages being easier to code, use, and access.

These new semantic elements, along with other standards like WAI-ARIA and Microdata (which we cover in Appendix B and Appendix C respectively), help make our documents more accessible to both humans and machines—resulting in benefits for both accessibility and search engine optimization.

The semantic elements, in particular, have been designed with the dynamic web in mind, with a particular focus on making pages more modular and portable. We'll go into more detail on this in later chapters.

Finally, the APIs associated with HTML5 help improve on a number of techniques that web developers have been using for years. Many common tasks are now simplified, putting more power in developers' hands. Furthermore, the introduction of HTML5-based audio and video means there will be less dependence on third-party software and plugins when publishing rich media content on the Web.

Overall, there is good reason to start looking into HTML5's new features and APIs, and we'll discuss more of those reasons as we go through this book.

What is CSS3?

Another separate—but no less important—part of creating web pages is Cascading Style Sheets (CSS). As you probably know, CSS is a style language that describes how HTML markup is presented or styled. CSS3 is the latest version of the CSS specification. The term "CSS3" is not just a reference to the new features in CSS, but the third level in the progress of the CSS specification.[5]

CSS3 contains just about everything that's included in CSS2.1 (the previous version of the spec). It also adds new features to help developers solve a number of problems without the need for non-semantic markup, complex scripting, or extra images.

[5] http://www.w3.org/Style/CSS/current-work.en.html