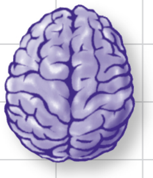
A Brain-Friendly Guide

Head First HTML5 Programming

A learner's guide to building web apps with JavaScript

Learn the secrets of the HTML5 guru

Discover why everything your friends know about video is probably wrong



Load HTML5 and JavaScript straight into your brain

Free Sampler

Avo
embarrassin
browse
support issue

Watch out for common browser pitfalls

O'REILLY®

Eric Freeman & Elisabeth Robson

O'Reilly Ebooks—Your bookshelf on your devices!

When you buy an ebook through oreilly.com you get lifetime access to the book, and whenever possible we provide it to you in five, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, Android .apk, and DAISY—that you can use on the devices of your choice. Our ebook files are fully searchable, and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at ebooks.oreilly.com

You can also purchase O'Reilly ebooks through the iBookstore, the Android Marketplace, and Amazon.com.

O'REILLY®

Spreading the knowledge of innovators

oreilly.com

Head First HTML5 Programming

by Eric Freeman and Elisabeth Robson

Copyright © 2011 Eric Freeman and Elisabeth Robson. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Series Creators:	Kathy Sierra, Bert Bates
Editor:	Courtney Nash
Design Editor:	Louise Barr
Cover Designer:	Karen Montgomery
Production Editor:	Kristen Borg
Indexer:	Ellen Troutman
Proofreader:	Nancy Reinhardt

Printing History:

October 2011: First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The *Head First* series designations, *Head First HTML5 Programming*, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and the authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

No gumballs were harmed in the making of this book.

ISBN: 978-1-449-39054-9

[M]

Table of Contents (Summary)

Table of Contents (the real thing)

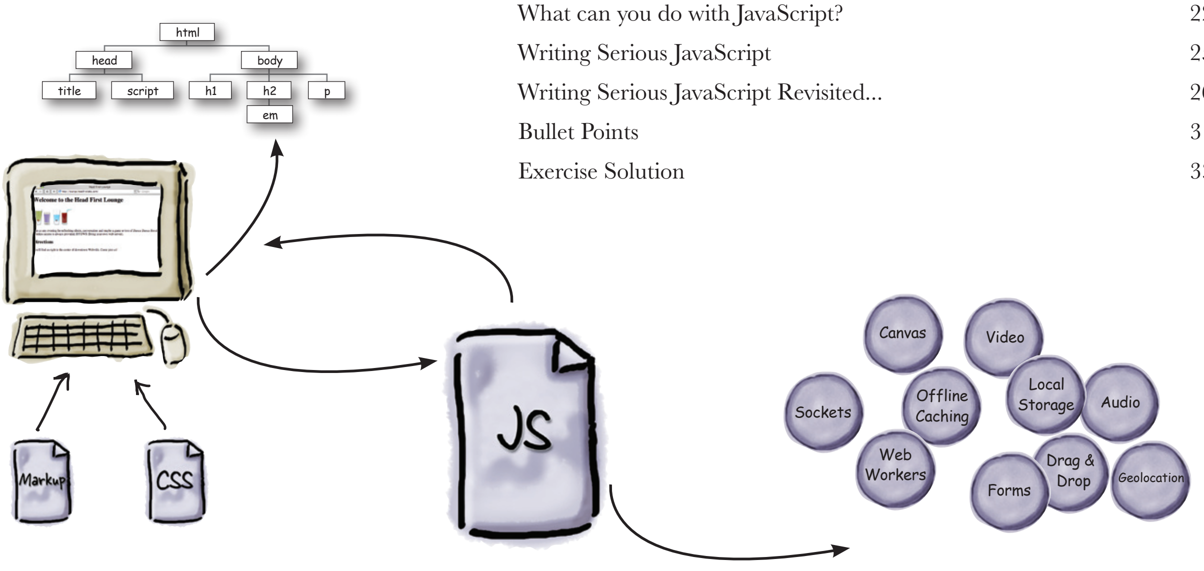
Intro

Your brain on HTML5 Programming. Here *you* are trying to *learn* something, while here your *brain* is doing you a favor by making sure the learning doesn't *stick*. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how *do* you trick your brain into thinking that your life depends on knowing HTML5 and JavaScript?

1 getting to know HTML5

Welcome to Webville

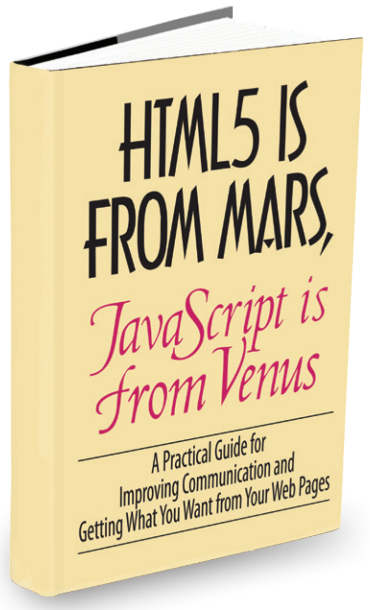
HTML has been on a wild ride. Sure, HTML started as a mere markup language, but more recently HTML's put on some major muscle. Now we've got a language tuned for building true web applications with local storage, 2D drawing, offline support, sockets and threads, and more. The story of HTML wasn't always pretty, and it's full of drama (we'll get to all that), but in this chapter, we're first going to go on a quick joyride through Webville to get sense for everything that goes into "HTML5." Come on, hop in, we're headed to Webville, and we're going to start by going from zero to HTML5 in 3.8 pages (flat).



2 introducing JavaScript and the DOM

A Little Code

JavaScript is going to take you to new places. You already know all about HTML markup (otherwise known as *structure*) and you know all about CSS style (otherwise known as *presentation*), but what you've been missing is JavaScript (otherwise known as *behavior*). If all you know about are structure and presentation, sure, you can create some great-looking pages, but they're still *just pages*. When you add behavior with JavaScript, you can create an interactive experience; or, even better, you can create full blown web applications. Get ready to add the most interesting and versatile skill in your web toolkit: JavaScript and programming!


HTML5 IS FROM MARS,
JavaScript is from Venus
A Practical Guide for Improving Communication and Getting What You Want from Your Web Pages

events, handlers and all that jazz

3 A Little Interactivity

You still haven't reached out to touch your user. You've learned the basics of JavaScript but can you get interactive with your users? When pages respond to user input, they aren't just documents anymore, they're living, reacting applications. In this chapter you're going to learn how to handle one form of user input (excuse the pun), and wire up an old-fashioned HTML <form> element to actual code. It might sound dangerous, but it's also powerful. Strap yourself in, this is a fast moving to-the-point-chapter where we go from zero to interactive app in no time.

4 javascript functions and objects

Serious JavaScript

Can you call yourself a scripter yet? Probably—you already know your way around a lot of JavaScript, but who wants to be a scripter when you can be a programmer? It's time to get serious and take it up a notch—it's time you learn about **functions** and **objects**. They're the key to writing code that is more powerful, better organized and more maintainable. They're also heavily used across HTML5 JavaScript APIs, so the better you understand them the faster you can jump into a new API and start ruling with it. Strap in, this chapter is going to require your undivided attention...

5 making your html location aware: Geolocation

Wherever you go, there you are. And sometimes knowing where you are makes all the difference (especially to a web app). In this chapter we're going to show you how to create web pages that are **location aware**—sometimes you'll be able to pin point your users down to the corner they're standing on, and sometimes you'll only be able to determine the area of town they're in (but you'll still know the town!). Heck, sometimes you won't be able to determine anything about their location, which could be for technical reasons, or just because they don't want you being so nosy. Go figure. In any case, in this chapter we're going to explore a JavaScript API: Geolocation. Grab the best location-aware device you have (even if it's your desktop PC), and let's get started.

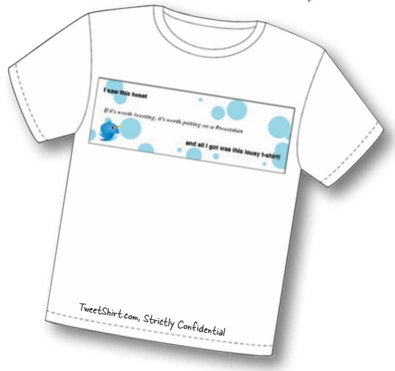
6 talking to the web
Extroverted Apps

You've been sitting in your page for too long. It's time to get out a little, to talk to web services, to gather data and to bring it all back so you can build better experiences mixing all that great data together. That's a big part of writing modern HTML5 applications, but to do that you've got to *know how* to talk to web services. In this chapter we're going to do just that, and incorporate some data from a real web service right in your page. And, after you've learned how to do that you'll be able to reach out and touch any web service you want. We'll even fill you in on the hippest new lingo you should use when talking to web services. So, come on, you're going to use some more APIs, the communications APIs.

Watch out for the cliffhanger in this chapter!

7 bringing out your inner artist: The Canvas

HTML's been liberated from being just a "markup" language. With HTML5's new canvas element you've got the power to create, manipulate and destroy *pixels*, right in your own hands. In this chapter we'll use the canvas element to bring out your inner artist—no more talk about HTML being all semantics and no presentation; with canvas we're going to paint and draw with color. Now it's *all* about presentation. We'll tackle how to place a canvas in your pages, how to draw text and graphics (using JavaScript of course), and even how to handle browsers that don't support the canvas element. And canvas isn't just a one-hit wonder; you're going to be seeing a lot more of canvas in other chapters in this book.


A new HTML5 startup is just waiting for you to get it off the ground!
I saw this tweet
If it's worth tweeting, it's worth putting on a #tweetshirt
and all I got was this lousy t-shirt!
TweetShirt.com, Strictly Confidential

8 not your father's tv

Video... with special guest star "Canvas"

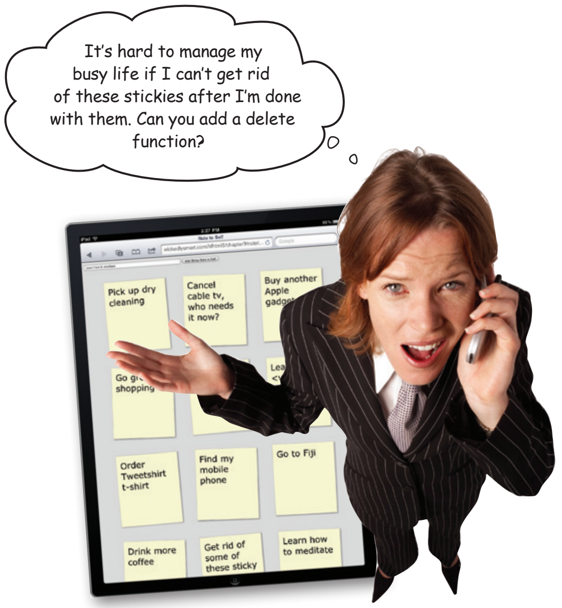
We don't need no plug-in. After all, video is now a first-class member of the HTML family—just throw a <video> element in your page and you've got instant video, even across most devices. But video is *far more* than *just an element*, it's also a JavaScript API that allows us to control playback, create our own custom video interfaces and integrate video with the rest of HTML in totally new ways. Speaking of *integration*... remember there's that *video and canvas connection* we've been talking about—you're going to see that putting video and canvas together gives us a powerful new way to *process video* in real time. In this chapter we're going to start by getting video up and running in a page and then we'll put the JavaScript API through its paces. Come on, you're going to be amazed what you can do with a little markup, JavaScript and video & canvas.

Tune in to Webville TV...

9 storing things locally

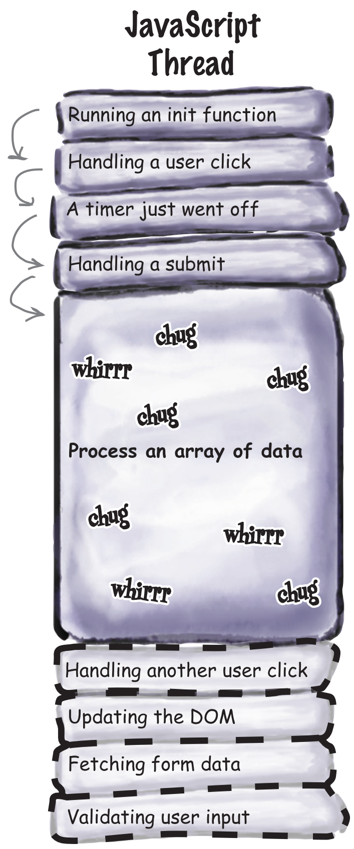
Web Storage

Tired of stuffing your client data into that tiny ~~closet~~ cookie? That was fun in the 90s, but we've got much bigger needs today with web apps. What if we said we could get you five megabytes on every user's browser? You'd probably look at us like we were trying to sell you a bridge in Brooklyn. Well, there's no need to be skeptical—the HTML5 Web storage API does just that! In this chapter we're going to take you through everything you need to store any object locally on your user's device and to make use of it in your web experience.


It's hard to manage my busy life if I can't get rid of these stickies after I'm done with them. Can you add a delete function?
Pick up dry cleaning
Cancel cable tv, who needs it now?
Buy another Apple gadget
Go grocery shopping
Order Tweetshirt t-shirt
Find my mobile phone
Go to Fiji
Drink more coffee
Get rid of some of these sticky
Learn how to meditate

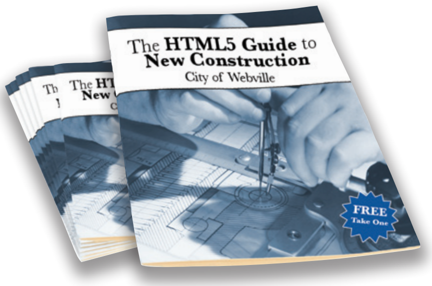
10 putting javascript to work
Web Workers

Slow script—do you want to continue running it? If you've spent enough time with JavaScript or browsing the web you've probably seen the "slow script" message. And, with all those multicore processors sitting in your new machine how could a script be running *too slow*? It's because JavaScript can only do one thing at a time. But, with HTML5 and Web Workers, *all that changes*. You've now got the ability to spawn *your own* JavaScript workers to get more work done. Whether you're just trying to design a more responsive app, or you just want to max out your machine's CPU, Web Workers are here to help. Put your JavaScript manager's hat on, let's get some workers cracking!



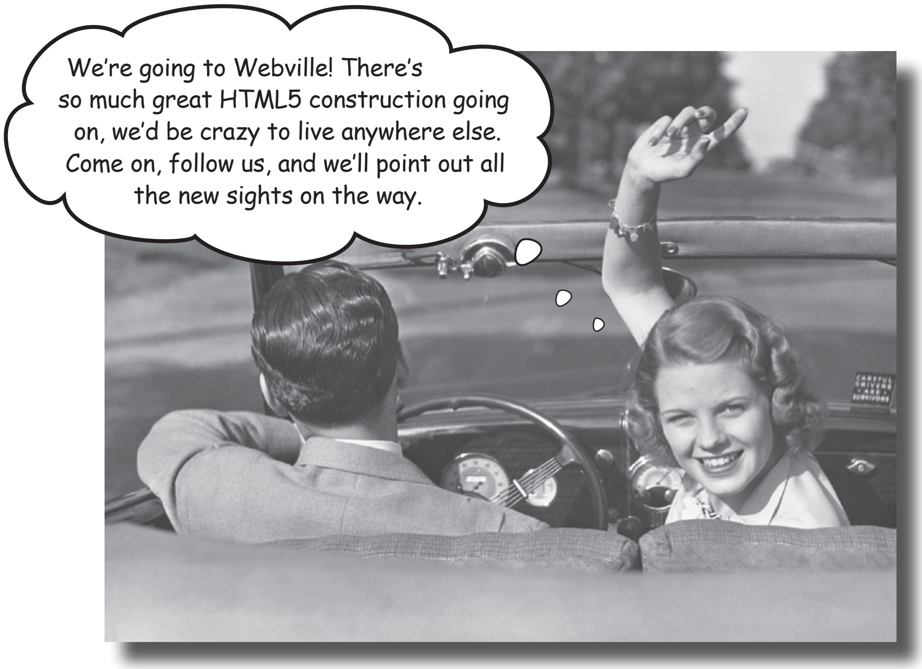
appendix: leftovers

We covered a lot of ground, and you're almost finished with this book. We'll miss you, but before we let you go, we wouldn't feel right about sending you out into the world without a little more preparation. We can't possibly fit everything you'll need to know into this relatively small chapter. Actually, we *did* originally include everything you need to know about HTML5 (not already covered by the other chapters), by reducing the type point size to .00004. It all fit, but nobody could read it. So, we threw most of it away, and kept the best bits for this Top Ten appendix.



1 getting to know html5

Welcome to Webville


We're going to Webville! There's so much great HTML5 construction going on, we'd be crazy to live anywhere else. Come on, follow us, and we'll point out all the new sights on the way.

HTML has been on a wild ride. Sure, HTML started as a mere markup language, but more recently HTML's put on some major muscle. Now we've got a language tuned for building true web applications with local storage, 2D drawing, offline support, sockets and threads, and more. The story of HTML wasn't always pretty, and it's full of drama (we'll get to all that), but in this chapter, we're first going to go on a quick joyride through Webville to get sense for everything that goes into "HTML5." Come on, hop in, we're headed to Webville, and we're going to start by going from zero to HTML5 in 3.8 pages (flat).

Heads up: XHTML received a "Dear John" letter in 2009 and we'll be visiting XHTML later in the "Where are they now" segment.

Upgrade to **HTML5** TODAY! Why wait? Use my

HTML5-o-Matic

and do it in only **THREE EASY STEPS**

Step right up! For a limited time we'll take that grungy old HTML page of yours and in **JUST THREE EASY STEPS** upgrade it to HTML5.

Could it really be that easy?
You betcha; in fact we've got a demonstration already prepared for you.

Check out this tired, worn out, seen-better-days HTML; we're going to turn it into HTML5 right before your very eyes:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Head First Lounge</title>
    <link type="text/css" rel="stylesheet" href="lounge.css">
    <script type="text/javascript" src="lounge.js"></script>
  </head>
  <body>
    <h1>Welcome to Head First Lounge</h1>
    <p>
      <img src="drinks.gif" alt="Drinks">
    </p>
    <p>
      Join us any evening for refreshing <a href="elixirs.html">elixirs</a>,
      conversation and maybe a game or two of Tap Tap Revolution.
      Wireless access is always provided; BYOWS (Bring Your Own Web Server).
    </p>
  </body>
</html>
```

This is all just normal HTML 4.01 from the Head First Lounge, which you might remember from Head First HTML (and if not, don't worry, you don't need to).

BRAIN POWER **Look how easy it is to write HTML5**

Get your feet wet by reviewing this HTML, which is written in HTML 4.01 (the previous version), not HTML5. Carefully look at each line and refresh your memory of what each part does. Feel free to make notes right on the page. We'll look at how to transition this to HTML5 over the next few pages.

Sharpen your pencil

After taking a careful look at the HTML on the page 2, can you see any markup that might change with HTML5? Or that you'd want to change? We'll point out one for you: the **doctype** definition:

This is the doctype for "html", we're in the right place!

This just means this standard is publicly available.

This part says we're using HTML version 4.01 and that this markup is written in ENglish.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
      "http://www.w3.org/TR/html4/strict.dtd">
```

This points to a file that identifies this standard.

Remember, the **doctype** definition belongs at the top of your HTML file and tells the browser the type of your document, in this case, HTML 4.01. By using a doctype the browser is able to be more precise in the way it interprets and renders your pages. Using a doctype is highly recommended.

So, using your deductive powers, what do you think the **doctype** definition for HTML5 will look like? Write it here (you can refer back to your answer when we cover this in a bit):

....

....

....

....

....

Your answer goes here.

Introducing the HTML5-o-Matic, update your HTML now!

STEP 1

Step 1 is going to amaze you: follow along, we're going to start at the top of the Head First Lounge HTML and update the doctype to give it that new HTML5 shine.

Here's the old HTML 4.01 version of the doctype:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
```

Now you might have guessed that we're going to replace every mention of "4" with "5" in the doctype, right? Oh, no. Here's the amazing part: the new doctype for HTML5 is simply:

```
<!doctype html>
```

Apologies to the crew that got the 4.01 doctype tattoo to remember it.

No more Googling to remember what the doctype looks like, or copying and pasting from another file, this doctype is so simple you can just remember it.

But, wait, there's more...

Not only is this the doctype for HTML5, it's the doctype for *every future version* of HTML. In other words, it's never going to change again. Not only that, it will work in your older browsers too.

The W3C HTML Standards guys have promised us they really mean it this time.

STEP 2

If you're a fan of the *Extreme Makeovers* or *The Biggest Loser* television shows, you're going to love Step 2. In this step we have the content meta tag... here, check out the before/after pictures:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

BEFORE (HTML 4)

```
<meta charset="utf-8">
```

AFTER (HTML5)

Yes, the new meta tag ~~has lost a lot of weight~~ is much simpler. When you specify the meta tag in HTML5, just supply the tag along with a character encoding. Believe it or not, all browsers (old and new) already understand this meta description, so you can use it on any page and it *just works.*

And now for Step 3, the step that brings it all home. Here we're also going to focus on the <head> element and upgrade the link tag. Here's what we have now: a link of type text/css that points to a stylesheet:

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

Old skool

To upgrade this for HTML5, we just need to remove the type attribute. Why? Because CSS has been declared the standard, and default, style for HTML5. So, after we remove the type attribute, the new link looks like this:

```
<link rel="stylesheet" href="lounge.css">
```

HTML5

And, because you acted fast, we've got a special bonus for you. We're going to make your life even easier by simplifying the script tag. With HTML5, JavaScript is now the standard and default scripting language, so you can remove the type attribute from your script tags too. Here's what the new script tag looks like without the type attribute:

```
<script src="lounge.js"></script>
```

Don't worry if you don't know a lot about the script tag yet, we'll get there...

Or if you have some inline code, you can just write your script like this:

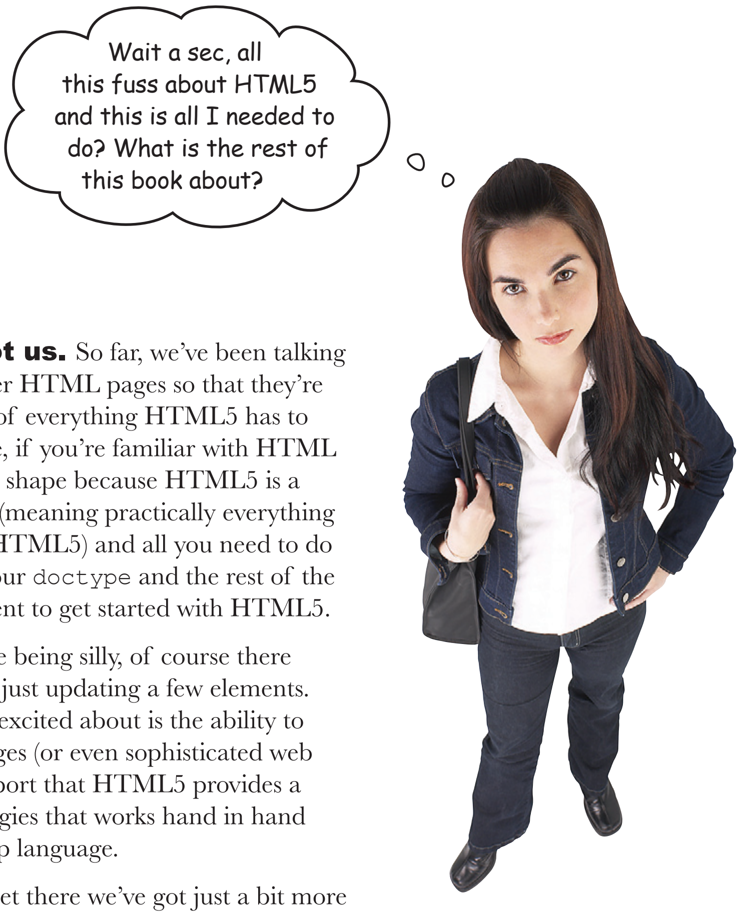
```
<script>

   var youRock = true;

</script>
```

All your JavaScript goes here.

We'll talk more about JavaScript in a bit.

Congratulations, you're now certified to upgrade any HTML to HTML5!

As a trained HTML5-o-Matic user, you've got the tools you need to take any valid HTML page and to update it to HTML5. Now it's time to put your certification into practice!


Wait a sec, all this fuss about HTML5 and this is all I needed to do? What is the rest of this book about?

Okay, okay, you got us. So far, we've been talking about updating your older HTML pages so that they're ready to take advantage of everything HTML5 has to offer. And as you can see, if you're familiar with HTML 4.01, then you're in great shape because HTML5 is a superset of HTML 4.01 (meaning practically everything in it is still supported in HTML5) and all you need to do is know how to specify your `doctype` and the rest of the tags in the `<head>` element to get started with HTML5.

But, you're right, we were being silly, of course there is more to HTML5 than just updating a few elements. In fact, what everyone is excited about is the ability to build rich, interactive pages (or even sophisticated web applications), and to support that HTML5 provides a whole family of technologies that works hand in hand with the HTML5 markup language.

But hang on; before we get there we've got just a bit more work to do to make sure we're ready with our markup.

Sharpen your pencil

You're closer to HTML5 markup than you think!

Here's some old skool HTML that needs updating. Work through the HTML5-o-Matic process and update this HTML to HTML5. Go ahead and scribble in the book, scratch out the existing markup code, and add any new markup code you need to. We've helped a little by highlighting the areas that need to change.

When you're done, type it in (or grab the exercise files and make your changes if you prefer), load this in your browser, sit back and enjoy your first HTML5. Oh, and you'll find our answers on the next page.

To download all the code and sample files for this book, please visit http://wickedlysmart.com/hfhtml5.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Head First Lounge</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <link type="text/css" rel="stylesheet" href="lounge.css">
    <script type="text/javascript" src="lounge.js"></script>
  </head>
  <body>
    <h1>Welcome to Head First Lounge</h1>
    <p>
      <img src="drinks.gif" alt="Drinks">
    </p>
    <p>
      Join us any evening for refreshing <a href="elixirs.html">elixirs</a>,
      conversation and maybe a game or two of Tap Tap Revolution.
      Wireless access is always provided;  BYOWS (Bring Your Own Web Server).
    </p>
  </body>
</html>
```

Sharpen your pencil Solution

You're closer to HTML5 markup than you think!

Here's some old skool HTML that needs updating. Work through the HTML5-o-Matic process and update this HTML to HTML5. Go ahead and scribble in the book, scratch out the existing markup code, and add any new markup code you need to. We've helped a little by highlighting the areas that need to change.

Here's our solution.

Here's the updated code:

```
<!doctype html>
<html>
  <head>
    <title>Head First Lounge</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lounge.css">
    <script src="lounge.js"></script>
  </head>
  <body>
    <h1>Welcome to Head First Lounge</h1>
    <p>
      <img src="drinks.gif" alt="Drinks">
    </p>
    <p>
      Join us any evening for refreshing <a href="elixirs.html">elixirs</a>,
      conversation and maybe a game or two of Tap Tap Revolution.
      Wireless access is always provided;  BYOWS (Bring Your Own Web Server).
    </p>
  </body>
</html>
```

The doctype...

Here are the four lines we changed to make our Head First Lounge web page officially HTML5.

... the meta tag...

... the link tag...

... and the script tag.

Don't believe us? Try http://validator.w3.org/ and you'll see – it validates as HTML5. For real!

there are no Dumb Questions

Q: **How does this work on the old browsers? Like the new doctype, meta, and so on... somehow the older browsers work with this new syntax?**

A: Yes, through a bit of cleverness and luck. Take the type attributes on the link and script tags; now, it makes sense to get rid of this attribute with HTML5 because CSS and JavaScript are now the standards (and certainly are the default technologies for style and scripting). But as it turns out, the browsers already assumed the defaults of CSS and JavaScript. So the stars aligned and the new markup standard just happens to have been supported in the browser for years. The same is true of the doctype and the meta tag.

Q: **What about the new doctype, it seems too simple now; it doesn't even have a version or DTD.**

A: Yes, it does seem a little magical that after years of using complex doctypes we can now just simplify it to "we're using HTML." Here's what happened: HTML used to be based on a standard called SGML, and that standard required both the complex form of the doctype and the DTD. The new standard has moved away from SGML as a way to simplify HTML language and make it more flexible. So, we don't need the complex form anymore. Further, as we said above, there is some luck here in that almost all browsers just look for HTML in the doctype to ensure they are parsing an HTML document.

Q: **Were you joking about it never changing again? I thought the versioning was really important for browsers. Why not use `<!doctype html5>`? It's not like there isn't going to be an HTML6 too. Right?**

A: The use of the doctype evolved with browser makers using the doctype to tell their browsers to render things in their own "standards mode." Now that we have much more of a true standard, the HTML5 doctype tells any browser that this document is standard HTML, be that version 5, 6 or whatever.

Q: **Well, I assume different browsers are going to have different capabilities at any one time. How do I handle that?**

A: True, especially until HTML5 is 100 percent supported. We'll cover both of these points in the chapter and throughout the book.

Q: **Why does this even matter? I just typed a page in without a doctype and meta tag and it worked just fine. Why do I need to worry if this stuff is totally correct?**

A: Yes, browsers are great at overlooking small errors in HTML files. But by including the correct doctype and meta tags, you'll make sure browsers know exactly what you want, rather than having to guess. Plus, for people using older browsers, the new doctype means they'll use standards mode, which is what you want. Remember, standards mode is a mode where the browser assumes you're writing HTML that conforms to a standard, so it uses those rules to interpret your page. If you don't specify a doctype, some browsers may go into "quirks mode" and assume your web page is written for older browsers, when the standard wasn't quite up to snuff, and may interpret your page incorrectly (or assume it's just written incorrectly).

Q: **Whatever happened to XHTML? It seems like a few years ago that was the future.**

A: Yeah it was. Then flexibility won out over strict syntax, and in the process XHTML (XHTML 2, to be precise) died and HTML5 was born to be more accepting of the way people write web pages (and the way browsers render them). That said, don't worry, because knowing about XHTML is only going to make you a stronger author of HTML5 content (and you're going to appreciate HTML5 a whole lot more). And by the way, if you really love XML, there's still a way to write your HTML5 in strict form. More on that later...

Q: **What is UTF-8?**

A: UTF-8 is a character coding that has support for many alphabets, including non-western ones. You've probably seen other character sets used in the past, but UTF-8 is being promoted as the new standard. And it's way shorter and easier to remember than previous character encodings.

Relax

We don't expect you to know HTML5, yet.

If you've never had exposure to HTML5 before, that's okay, but you should have worked with HTML, and there are some basics you should know about like elements, tags, attributes, nesting, the difference between semantic markup and adding style, and so on.

If you aren't familiar with all these, we're going to make a small suggestion (and a shameless plug): there's another book that proceeds this one, *Head First HTML with CSS & XHTML*, and you should read it. And if you're somewhat familar with markup languages, you might want to skim it or use it as a reference while reading this book.

We've also put a small guide to HTML5 markup & CSS3 in the appendix. If you just want a quick overview of the new additions, have a quick read over them at the end of the book.

A Brain-Friendly Guide
Head First HTML with CSS & XHTML
O'REILLY
Elisabeth Freeman & Eric Freeman

HTML5 Exposed

This week's interview:
Confessions of the newest version of HTML

Head First: Welcome, HTML5. All the Web is buzzing about you. To us, you look a lot like HTML 4. Why is everyone so excited?

HTML5: Everyone's excited because I'm enabling a whole new generation of web applications and experiences.

Head First: Right, but again, why didn't HTML 4 or even the promise of "XHTML" do that?

HTML5: XHTML 2 was a dead-end. Everyone who wrote real web pages hated it. XHTML reinvented the way we write markup for a web page, and would have made all the pages already out there obsolete. I said, "Hey, wait a sec, I can do new things and embrace everything that is already out there." I mean, if something works, why reinvent the wheel? That's my philosophy.

Head First: It seems to be working. But you know, some of the standards guys are still saying that the Web would be better off following their "pure" standards.

HTML5: You know, I don't really care. I listen to the people out there writing real web pages—how are they using me, and how can I help? Second on my list are the developers creating the web browsers. And last on my list are the standards guys. I'll listen to them, but not if it disagrees with what real users are doing.

Head First: Why not?

HTML5: Because if the users and the browser-makers disagree with the standards guys, it is a moot point. Luckily the people working on the HTML5 spec totally agree with me, and that's our philosophy.

Head First: Back to the previous version of HTML, you've said you are a superset of HTML 4.01. That means you're backward-compatible, right? Does that mean you're going to have to keep handling all the bad designs of the past?

HTML5: I promise I'll do my best to handle anything from the past that is thrown at me. That said, it doesn't mean that is the way to treat me. I do want web page authors to be educated on the latest standard and use me in the best way possible. That way, they can really push me to my limits. But again, I won't totally fail, and I will display an old page to the best of my ability if it's not updated.

Head First: My next question is ...

HTML5: Hold on, hold on!!! All these questions about the past. We aren't talking about what is important here. As far as my markup is concerned, my personal mission is to embrace the Web as it is, add some new structured elements that make web author's lives easier, and to help all browser implementors support consistent semantics around my markup. But I'm really here to pitch my new purpose: web applica...

Head First: ...So sorry HTML5, that's all we have time for. Thanks, and we'll be sure to talk about anything you want in an upcoming interview.

HTML5: Argh, I hate when that happens!!!

Would the REAL HTML5 please stand up...

Okay, you've patiently humored us by sitting through our "HTML5-o-Matic" skit, and we're sure you've already guessed there's a lot more to HTML5 than that. The word on the street is that HTML5 removes the need for plug-ins, can be used for everything from simple pages to Quake-style games and is a whipped topping for desserts. HTML5 seems to be something different to everyone...

HTML5 is all about multimedia, getting rid of plug-ins and using the new native support for audio and video.

Actually, it's about rich Internet clients. Instead of building clients with plug-ins like Flash, now I can use canvas, transforms and JavaScript to make cool interfaces and animations.

The great thing about HTML5 is the client-side storage and caching functionality. Can you say offline access to the Web?

I'm excited because I can use web workers to make my JavaScript more efficient and my page feel more responsive.

There's also a ton of new stuff in CSS we can use with HTML5. Advanced selectors, animations, and yeah—drop shadows!

Don't forget mobile. I want to be able to write web pages that know where I am.

The good news is, HTML5 is all these things. When people talk about HTML5 they mean a *family of technologies* that, when combined, gives you a whole new palette for building web pages and applications.

How HTML5 really works...

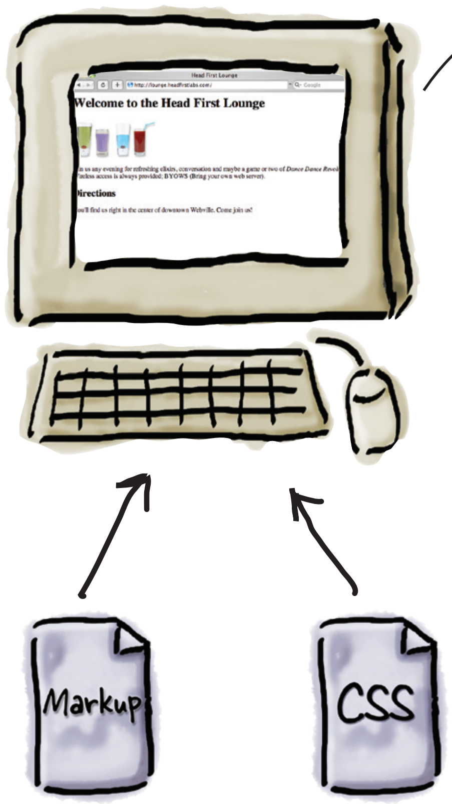
So we've said HTML5 is made up of a family of technologies, but what does that mean? Well you already know there's the HTML markup itself, which has been expanded to include some new elements; there are also lots of additions to CSS with CSS3 that give you even more power to style your pages. And then there's the turbo charger: *JavaScript*, and a whole new set of JavaScript APIs that are available to you.

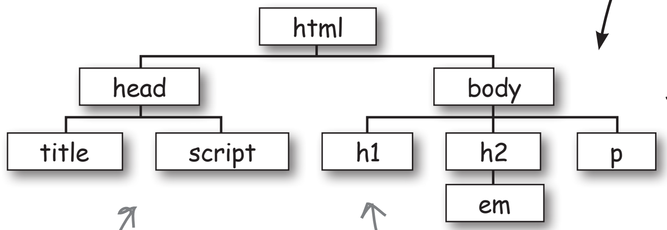
You'll find a nice Webville guide to the new HTML5 markup & CSS3 properties in the appendix.

Let's take a look behind the scenes and see how this all fits together:

1 The **browser** loads a document, which includes markup written in HTML and style written in CSS.

2 As the browser loads your page, it also creates an internal **model of your document** that contains all the elements of your HTML markup.





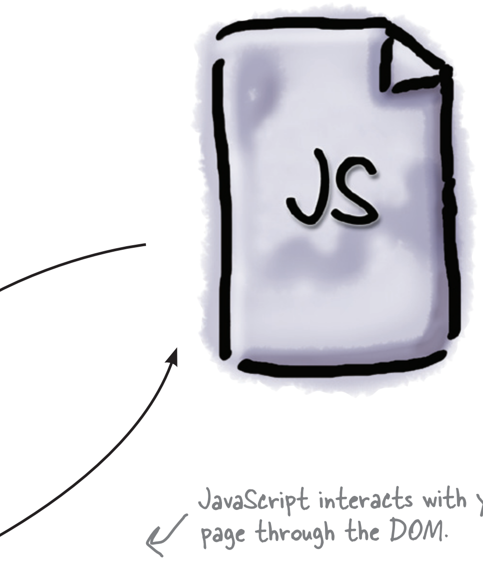
This is where it gets interesting; for each element in your HTML, the browser creates an object that represents it and places it in a tree-like structure with all the other elements...

...we call this tree the Document Object Model or the DOM for short. You'll be seeing a lot more of the DOM in this book because it plays a vital role in how we add behavior to pages with JavaScript (we'll get to that shortly, in Chapter 2).

The page's style (if it has any) comes from CSS3, which has been expanded from CSS2 to include many common idioms that are in use across the Web (like drop shadows and rounded corner borders).

With HTML5 the markup has some improvements, as you've seen with the tags in the <head> element, and there are some additional elements you can use (we'll see a few in this book).

Behind the Scenes



3 While the browser is loading your page it's also loading your **JavaScript code,** which typically begins executing just after the page loads.

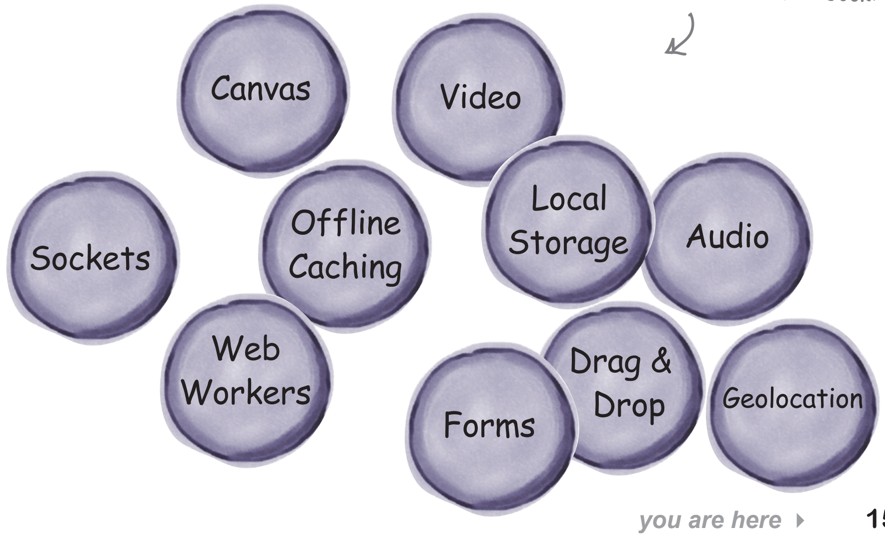
Using JavaScript, you can interact with your page by manipulating the DOM, react to user or browser-generated events, or make use of all the new APIs.

JavaScript interacts with your page through the DOM.

4 The **APIs** give you access to audio, video, 2D drawing with the canvas, local storage and a bunch of other great technologies needed to build apps. And remember, to make use of all these APIs, we need JavaScript.

APIs, otherwise known as Application Programming Interfaces, expose a set of objects, methods, and properties that we can use to access all the functionality of these technologies. We'll be covering many of these APIs in this book.

Meet the JavaScript APIs



WHO DOES WHAT?

We've already talked about the "family of technologies" so much we feel like they're, well, family. But then again we really haven't gotten to know them yet, so isn't it about time? You'll find most of the family below, so go ahead mingle, see if you can figure out who is who. We've gone ahead and figured one out for you. *And don't worry, we know this is your first time meeting the HTML5 family members, so the answers are at the end of the chapter.*

CSS3

Web Workers

Forms

Offline Web Apps

Audio & Video

New Markup

Local Storage

Canvas

Geolocation

Using me, you can draw right on your web page. With me, you can draw text, images, lines, circles, rectangles, patterns and gradients. I'll bring out your inner artist.

You might have used me in HTML 4 to enter information, but I'm even better in HTML5. I can require that you fill out fields, and I can more easily verify that you've typed an email, URL or phone number where you're supposed to.

You used to need a plug-in for us, but now we're first class members of the HTML family of elements. Wanna watch or listen to something? You need us.

We're here to help with the structure and semantic meaning of your page, including new ways of making sections, headers, footers and navigation in your pages.

I'm the most stylish one in the family. You've probably used me before, but did you know I can now animate your elements, give them great rounded corners and even drop shadows?

Use me as a bit of local storage in every user's browser. Need to store a few preferences, some shopping cart items, or maybe even stash a huge cache for efficiency? I'm your API.

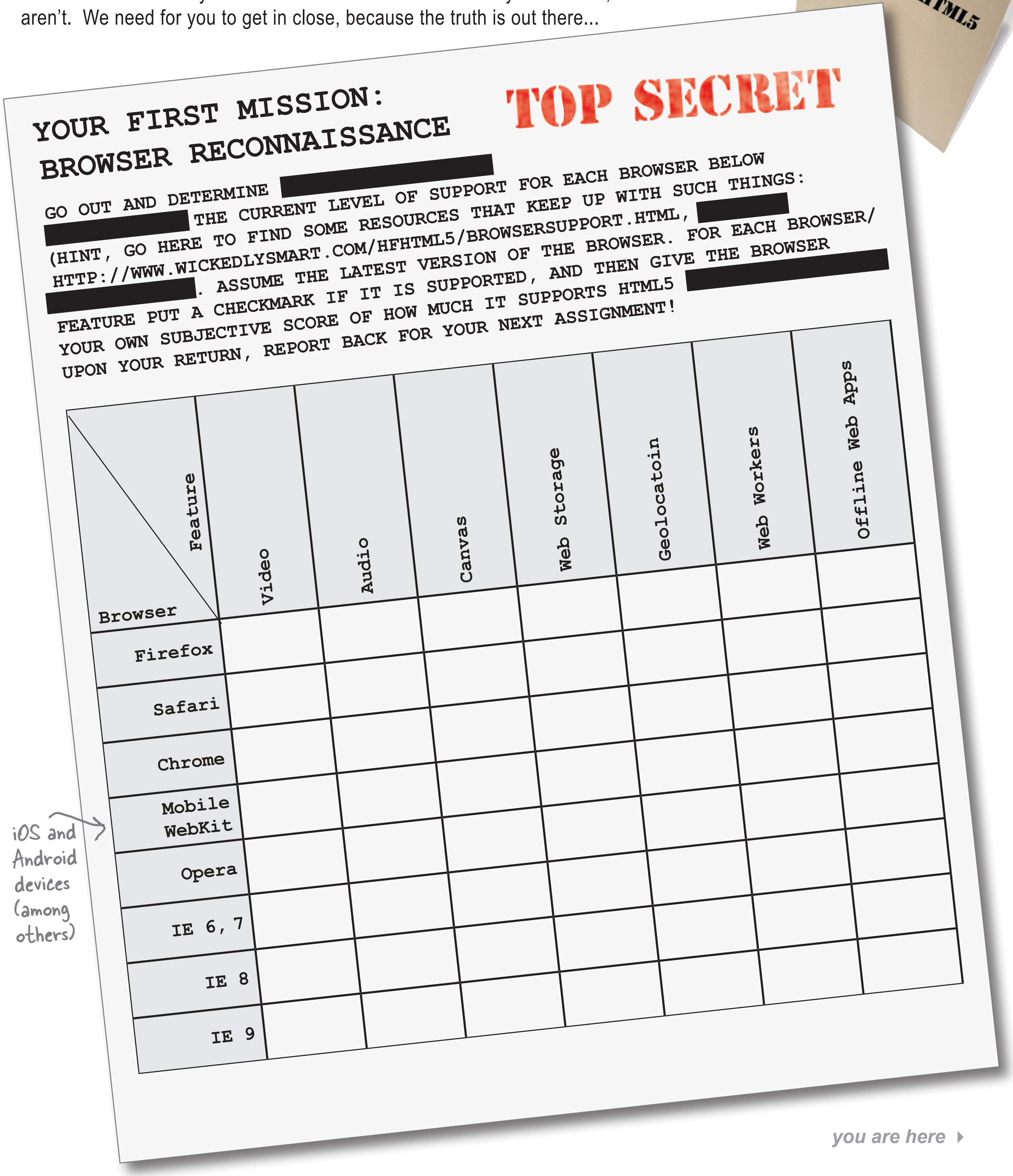
Need applications that work even when you aren't connected to the network? I can help.

I'm the API that can tell you where you are, and I play nice with Google maps.

You'll want me whenever you need several scripts running concurrently and in the background, so your user interface remains responsive.

YOUR MISSION...

...should you choose to accept it, is to do some reconnaissance on all the HTML browsers. We're sure you've heard some browsers are ready for HTML5, and some aren't. We need for you to get in close, because the truth is out there...



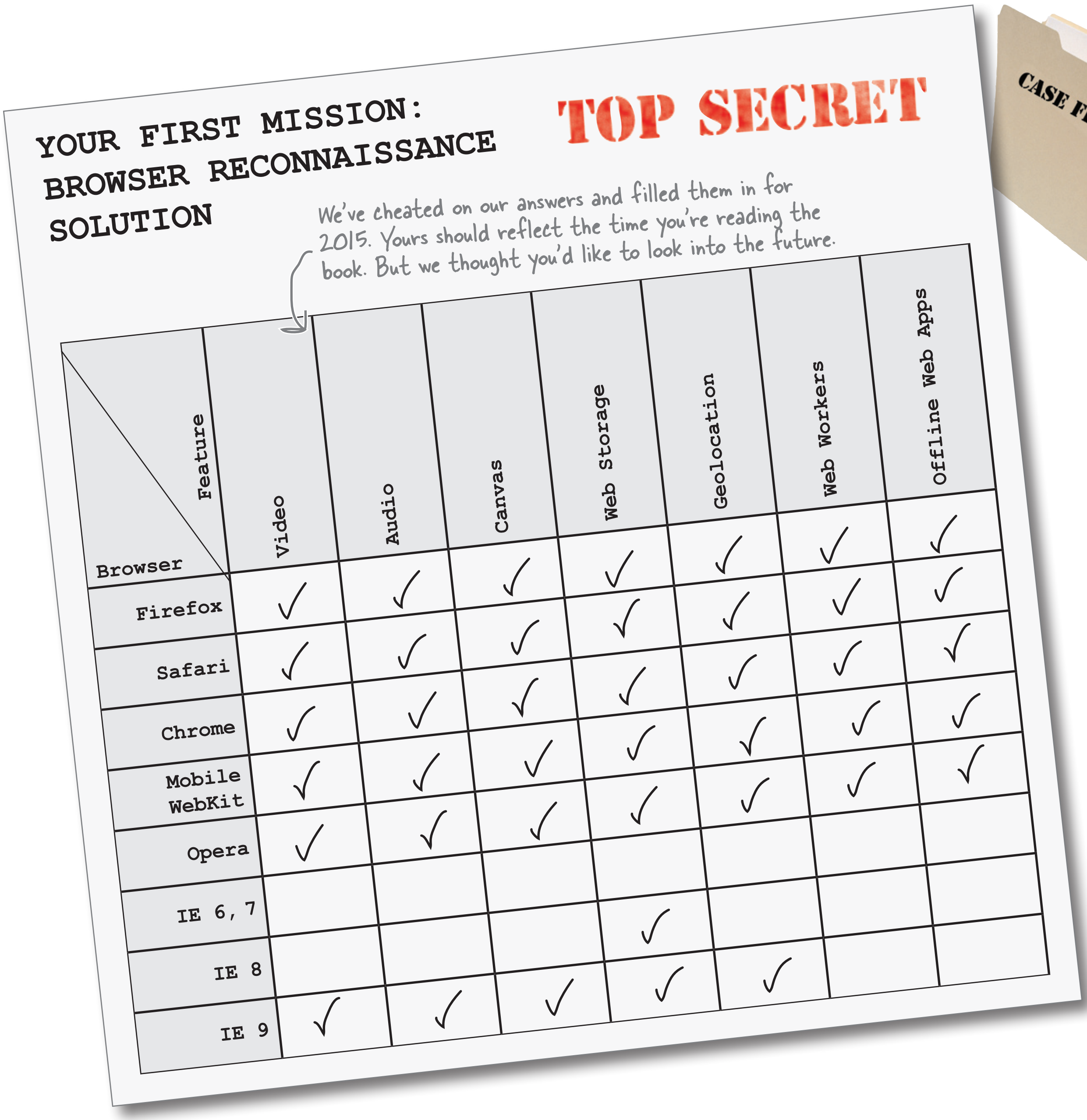
YOUR FIRST MISSION:
BROWSER RECONNAISSANCE

TOP SECRET

GO OUT AND DETERMINE [redacted]
[redacted] THE CURRENT LEVEL OF SUPPORT FOR EACH BROWSER BELOW
(HINT, GO HERE TO FIND SOME RESOURCES THAT KEEP UP WITH SUCH THINGS:
HTTP://WWW.WICKEDLYSMART.COM/HFHTML5/BROWSERSUPPORT.HTML, [redacted]
[redacted]. ASSUME THE LATEST VERSION OF THE BROWSER. FOR EACH BROWSER/
FEATURE PUT A CHECKMARK IF IT IS SUPPORTED, AND THEN GIVE THE BROWSER
YOUR OWN SUBJECTIVE SCORE OF HOW MUCH IT SUPPORTS HTML5 [redacted]
UPON YOUR RETURN, REPORT BACK FOR YOUR NEXT ASSIGNMENT!

Feature / Browser	Video	Audio	Canvas	Web Storage	Geolocatoin	Web Workers	Offline Web Apps
Firefox							
Safari							
Chrome							
Mobile WebKit							
Opera							
IE 6, 7							
IE 8							
IE 9							

iOS and Android devices (among others)

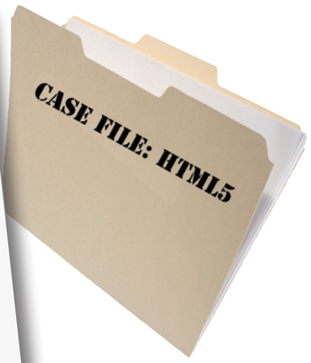


YOUR FIRST MISSION:
BROWSER RECONNAISSANCE
SOLUTION

TOP SECRET

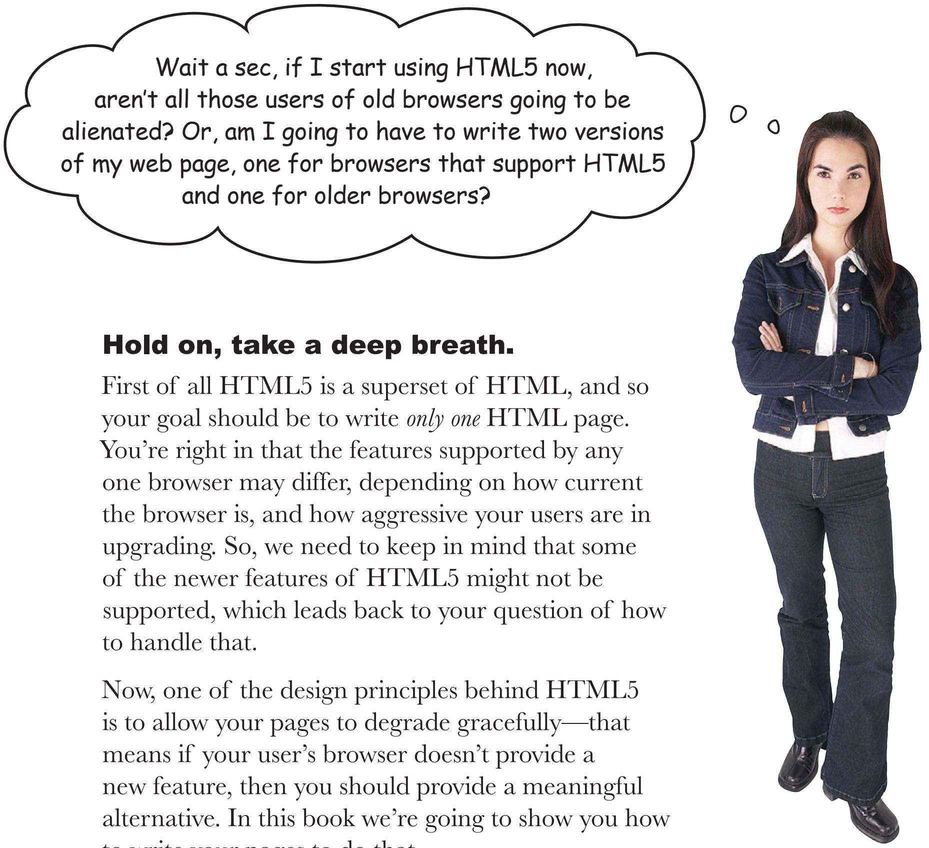
We've cheated on our answers and filled them in for 2015. Yours should reflect the time you're reading the book. But we thought you'd like to look into the future.

Browser \ Feature	Video	Audio	Canvas	Web Storage	Geolocation	Web Workers	Offline Web Apps
Firefox	✓	✓	✓	✓	✓	✓	✓
Safari	✓	✓	✓	✓	✓	✓	✓
Chrome	✓	✓	✓	✓	✓	✓	✓
Mobile WebKit	✓	✓	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓	✓	✓
IE 6, 7							
IE 8				✓			
IE 9	✓	✓	✓	✓	✓		



Even though it will be a while before the standard gets signed, sealed and delivered, you'll be using browsers that fully support HTML5 long before then. In fact, on modern browsers many features are already supported across the board. That's why it's a great idea to get started using HTML5 now. Plus if you start now, you'll be able to impress your friends and coworkers with all your cutting edge knowledge.

And get that raise sooner!


Wait a sec, if I start using HTML5 now, aren't all those users of old browsers going to be alienated? Or, am I going to have to write two versions of my web page, one for browsers that support HTML5 and one for older browsers?

Hold on, take a deep breath.

First of all HTML5 is a superset of HTML, and so your goal should be to write *only one* HTML page. You're right in that the features supported by any one browser may differ, depending on how current the browser is, and how aggressive your users are in upgrading. So, we need to keep in mind that some of the newer features of HTML5 might not be supported, which leads back to your question of how to handle that.

Now, one of the design principles behind HTML5 is to allow your pages to degrade gracefully—that means if your user's browser doesn't provide a new feature, then you should provide a meaningful alternative. In this book we're going to show you how to write your pages to do that.

But the good news is that all browsers are moving towards the HTML5 standard and related technologies (even the mobile browsers) and so over time graceful degradation will be more the exception than the rule (although you'll always want to do what you can to give your users a meaningful experience no matter what browser they're on).

there are no Dumb Questions

Q: I heard that the HTML5 Standard isn't going to be a final recommendation until 2022! Is that true? And, if so, why are we bothering?

A: The W3C is the standards body that formally recommends the HTML5 standard, and what you need to know about the W3C is that they are a conservative bunch, so conservative that they'd prefer to wait until a few generations of HTML5 browsers have come and gone before they give their signoff. That's okay; the standard should be wrapped up in the next couple years and the browser makers are well on to their way to implementing it. So, yes, it may be quite a while before HTML5 is a "final recommendation," but it's expected to be a stable standard by 2014, and for all practical purposes you should get going now on HTML5.

Q: What happens after HTML5 is final?

A: HTML6? We have no idea, but maybe whatever it is it will come with flying cars, rocket suits and dinner in a pill. Remember that even if we do adopt HTML6, the doctype won't change. Assuming the W3C keeps their promise and future versions of HTML remain backward-compatible, we'll be in good shape to take whatever comes next.

Q: Chrome, Safari, Firefox, a zillion mobile browsers...isn't the world just getting worse? How will we ever make sure our pages work on all these browsers?

A: While there is plenty of healthy competition in the marketplace for browsers (desktop and mobile), in reality many of these browsers are based on a few common HTML engines. For instance Chrome, Safari and the mobile browsers in the Android and iPhone are all based on WebKit, an open source browser engine. So, for the most part, your pages will work out of the gate on multiple browsers without a lot of effort on your part.

Q: Why not just use Flash to solve cross-browser issues?

A: Flash is a great tool for many applications and certainly on the desktop it is pervasive across operating systems and browsers. HTML5 and its family of technologies is trying to allow you to do with open standards many of the same things that Flash can do. So which way should you go? One thing to think about is the amount of investment going into HTML5 technologies within Google, Apple, Microsoft, and others. Over the long term, HTML5 is going to be a huge player, and in the mobile space it already is. So, while the choice is yours, and we're sure both are going to be around for a long time, the industry is heading towards open standards.

HTML Archaeology

We did some digging and found some code embedded in an HTML page. We're hoping you can help us crack the code to figure out what it means. Don't worry; we don't expect you to understand this code, we're just trying to get your brain warmed up with a little deductive reasoning...

```
<script>
    var walksLike = "duck";
    var soundsLike = document.getElementById("soundslike");
    if (walksLike == "dog") {
        soundsLike.innerHTML = "Woof! Woof!";
    } else if (walksLike == "duck") {
        soundsLike.innerHTML = "Quack, Quack";
    } else {
        soundsLike.innerHTML = "Crickets...";
    }
</script>
```

A hint: document represents the entire HTML page, and getElementById probably has something to do with HTML elements and ids.

I'm just sayin', if you're going to get serious about building web apps and using HTML5, you've got to have JavaScript chops.

We've gotta talk.

If you've been with us since *Head First HTML & CSS* (or, you've read this far into the book without repurposing it as firewood) we know you probably have a good understanding of using markup languages and stylesheets to create great looking web pages. Knowing those two technologies can you get a long way...

But, with HTML5 things are changing: web pages are becoming rich experiences (and full blown applications) that have behavior, are updated on the fly, and interact with the user. Building these kinds of pages requires a fair bit of programming and if you're going to write code for the browser, there's only one game in town: *JavaScript*.

Now, if you've programmed or written simple scripts before, you're going to be in good shape: JavaScript (despite some rumors) is a fantastic language and we'll take you through everything you need to know to write the applications in this book. If you haven't programmed before, we're going to do everything we can to take you along for the ride. In either case, one of the huge benefits of JavaScript is how accessible it is to new programmers.

We can't think of a better or more fun way to learn to program!

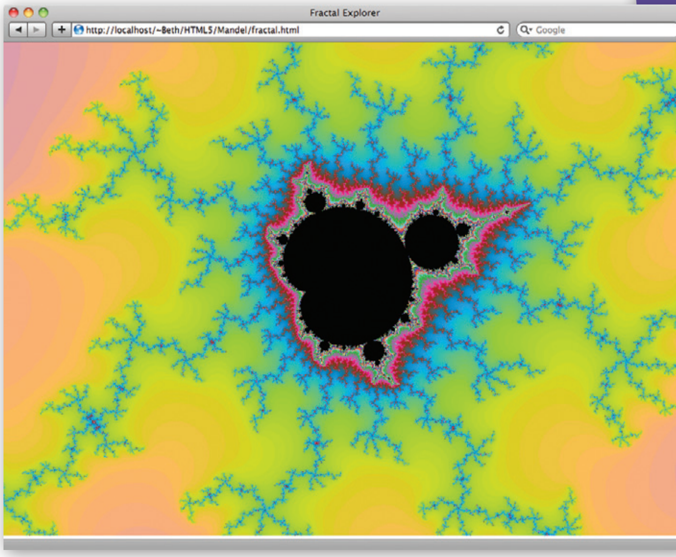
So, what now? Let's just briefly get introduced to a little JavaScript and then we'll really dive in deep in Chapter 2. In fact, for now, don't worry too much about getting every detail over the next few pages—we just want you to get a *feel for JavaScript*.

What can you do with JavaScript?

JavaScript opens up a whole new universe of expression and functionality to your web pages. Let's look at just a few things you might do with JavaScript and HTML5...

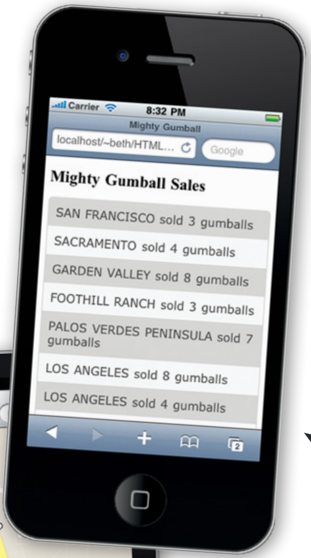
Interact with your pages in new ways that work for the desktop and mobile devices.

With HTML5 & JavaScript you can create a 2D drawable surface right in your page, no plug-ins required.



Use web workers to turbo-charge your JavaScript code and do some serious computation or make your app more responsive. You can even make better use of your user's multicore processor!

Make your pages location aware to know where your users are, show them what's nearby, take them on a scavenger hunt, give them directions, or to bring people with common interests together in the same area.



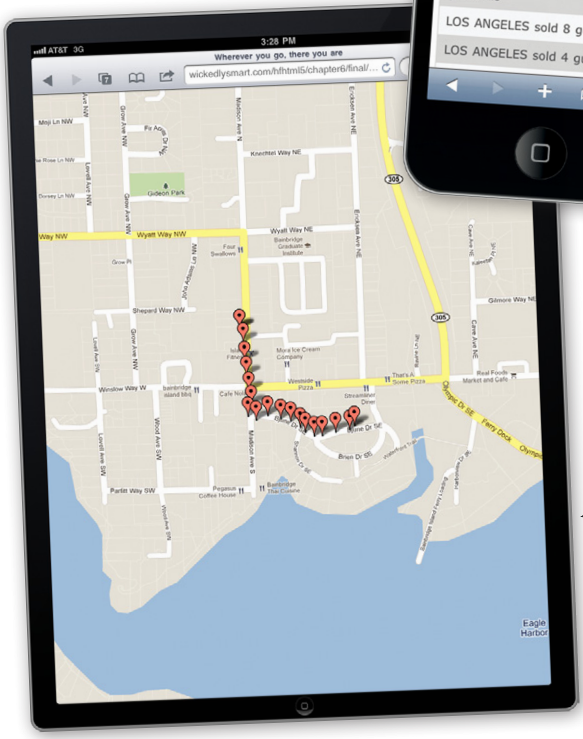
Mighty Gumball Sales

SAN FRANCISCO sold 3 gumballs
SACRAMENTO sold 4 gumballs
GARDEN VALLEY sold 8 gumballs
FOOTHILL RANCH sold 3 gumballs
PALOS VERDES PENINSULA sold 7 gumballs
LOS ANGELES sold 8 gumballs
LOS ANGELES sold 4 gumballs

Access any web service and bring that data back to your app, in near real time.

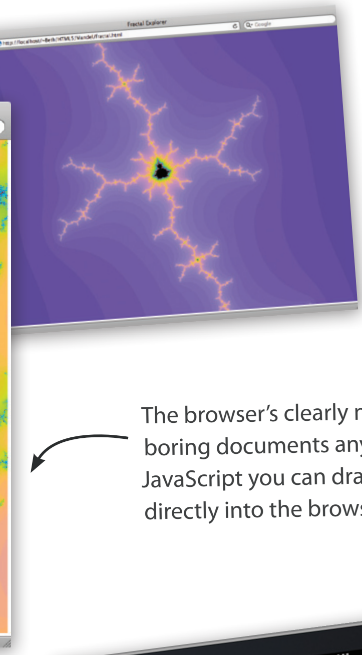
Cache data locally using browser storage to speed up mobile apps.

No need for special plug-ins to play video.


AT&T 3G
3:28 PM
Wherever you go, there you are
wickedlysmart.com/hfhtml5/chapter6/final/...
Eagle Harbor

Integrate your pages with Google Maps and even let your users track their movement in real time.

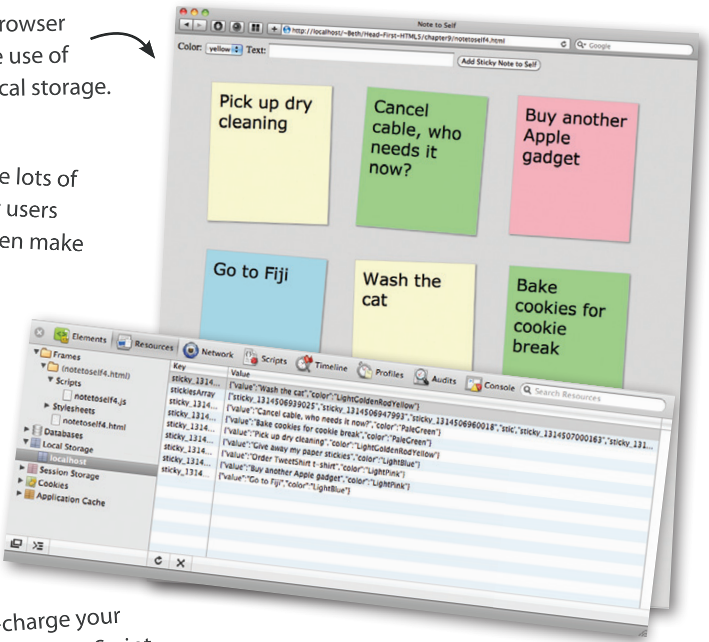
Create your own video playback controls using HTML and JavaScript.



The browser's clearly not just for boring documents anymore. With JavaScript you can draw pixels directly into the browser.

Say goodbye to browser cookies and make use of browser-based local storage.

Using JavaScript you can store lots of preferences and data for your users locally, in the browser, and even make it available for offline access.



Pick up dry cleaning

Cancel cable, who needs it now?

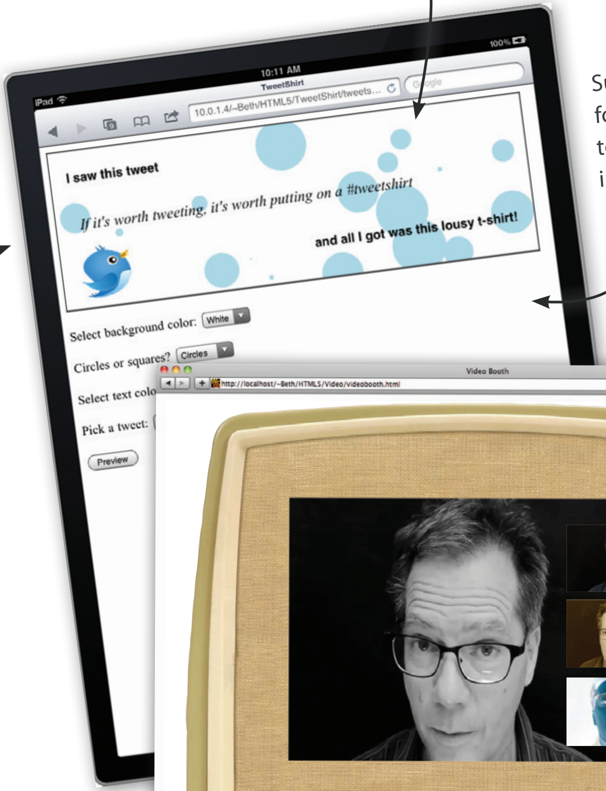
Buy another Apple gadget

Go to Fiji

Wash the cat

Bake cookies for cookie break

```
Key           Value
sticky_1314... {"value":"Wash the cat","color":"LightGoldenRodYellow"}
stickiesArray  ["sticky_1314506939025","sticky_1314506947993","sticky_1314506960018","stic","sticky_1314507000163","sticky_131...
sticky_1314... {"value":"Cancel cable, who needs it now?","color":"PaleGreen"}
sticky_1314... {"value":"Bake cookies for cookie break","color":"PaleGreen"}
sticky_1314... {"value":"Pick up dry cleaning","color":"LightGoldenRodYellow"}
sticky_1314... {"value":"Give away my paper stickies","color":"LightBlue"}
sticky_1314... {"value":"Order TweetShirt t-shirt","color":"LightPink"}
sticky_1314... {"value":"Buy another Apple gadget","color":"LightPink"}
sticky_1314... {"value":"Go to Fiji","color":"LightBlue"}
```



I saw this tweet

If it's worth tweeting, it's worth putting on a #tweetshirt

and all I got was this lousy t-shirt!

Select background color: White

Circles or squares? Circles

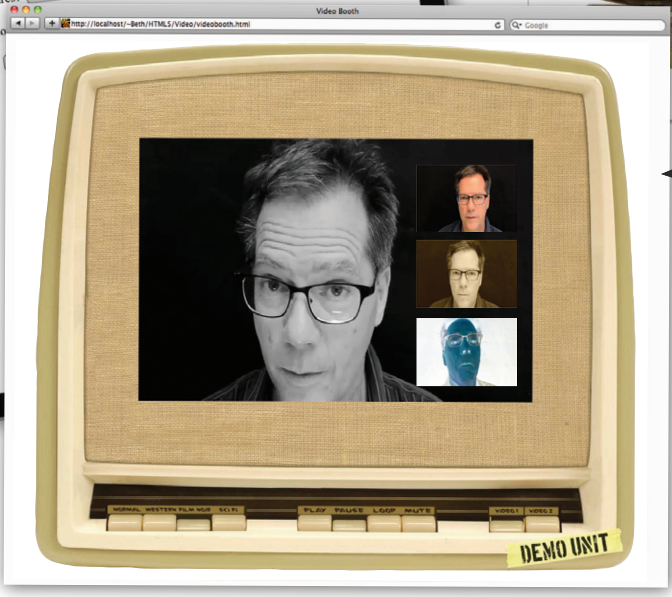
Select text colo

Pick a tweet:

Preview

Super-charge your forms with JavaScript to provide real interactivity.

Build complete video experiences that incorporate video in new ways.



Use the power of JavaScript to do full blown video processing in your browser. Create special effects and even directly manipulate video pixels.

You probably think we searched the Web far and wide to find the most exciting examples we could, right? Nope. All we did was take screenshots of the examples in the rest of this book. How's that for cool? So now that you're in Webville, it's time to learn the local lingo: JavaScript. Come on, let's get started.

JavaScript Exposed

This week's interview:
Confessions of a Scripting Language

Head First: Welcome, JavaScript. We're glad you could work us into your busy schedule. Let me just put it out there: HTML5 is becoming quite a celebrity—what's your take on this?

JavaScript: I'm not someone who seeks the limelight, I'm a behind the scenes kinda guy. That said, a lot of the credit going to HTML5 should be going to me.

Head First: Why do you say that?

JavaScript: There's a whole family of technologies that makes "HTML5" work, like the 2D canvas, local storage, web workers, that kind of thing. And the truth is, it takes me, JavaScript, to really make use of them. Sure, HTML5 gives you a place to hold the whole experience together and present it, but without me, you wouldn't have an interesting experience at all. That's okay, more power to HTML5; I'm just going to keep on doing my job.

Head First: What's your advice for new HTML5 authors?

JavaScript: That's easy. If you want to really master HTML5, spend your time on JavaScript and all the libraries that work with HTML5.

Head First: You know, you haven't always had the best reputation. I'll quote a review from 1998: "JavaScript is at best a half-baked, wimpy scripting language."

JavaScript: That hurts. I may not have started life in the clean, academic environment of many programming languages, but I've become one of the most widely used programming languages of all time, so I wouldn't discount me so quickly. Not only that, but enormous resources have been poured into making me robust and extremely fast. I'm at least 100 times faster than I was a decade ago.

Head First: That's impressive.

JavaScript: Oh, and if you haven't heard, the standards guys also just told me I'm now the default scripting language for HTML5. So, I'm here to stay. In fact, you don't even have to say "JavaScript" in your `<script>` tag anymore. So they may have called me wimpy in '98, but where are JScript, VBScript, Java Applets and all those failed attempts at browser languages now?

Head First: Well it certainly sounds like you are the key to creating great HTML5 experiences. You do have a reputation for being a confusing language.

JavaScript: I'm a very powerful language, despite some rumors, so you should really spend some time learning to use me well. On the other hand, I'm popular because I'm so easy to get up and running with. The best of both worlds, don't you think?

Head First: It sounds that way! Thanks, JavaScript, for joining us.

JavaScript: My pleasure, anytime.

Writing Serious JavaScript

With all this talk about JavaScript, we bet you're ready to jump in and see what it's all about. They don't call this *Head First* for nothing, we've got a super serious business application below that we're going to throw at you. For now, get started by going through the code to get a feel for it. Write down what you think each line does. Don't worry, we don't expect you to understand everything yet, but we bet you can make some really good guesses about what this code does. And, when you're done, turn the page and see how close you were...

Write your answers here.

Substitute your favorite drink here.

```
var drink = "Energy Drink";
var lyrics = "";
var cans = 99;

while (cans > 0) {
    lyrics = lyrics + cans + " cans of "
            + drink + " on the wall <br>";
    lyrics = lyrics + cans + " cans of "
            + drink + "<br>";
    lyrics = lyrics + "Take one down, pass it around,<br>";

    if (cans > 1) {
        lyrics = lyrics + (cans-1) + " cans of "
            + drink + " on the wall <br>";
    }

    else {
        lyrics = lyrics +  "No more cans of "
            + drink + " on the wall <br>";
    }
    cans = cans - 1;
}
document.write(lyrics);
```

Writing Serious JavaScript Revisited...

Walk through the code again and see if you were on the mark. At this point you just want to get a feel for the code; we'll be stepping through everything in detail soon enough.

```
var drink = "Energy Drink";

var lyrics = "";

var cans = 99;


while (cans > 0) {


    lyrics = lyrics + cans + " cans of "

               + drink + " on the wall <br>";

    lyrics = lyrics + cans + " cans of "

               + drink + "<br>";

    lyrics = lyrics + "Take one down, pass it around,<br>";

    if (cans > 1) {

        lyrics = lyrics + (cans-1) + " cans of "

                    + drink + " on the wall <br>";

    }

    else {

        lyrics = lyrics +  "No more cans of "

                    + drink + " on the wall <br>";

    }

    cans = cans - 1;

}


document.write(lyrics);
```

Declare a variable, and assign it a value of "Energy Drink".
Declare another variable and assign it empty string value.
Declare another variable and assign it a number value, 99.
This is a while loop. It says, while the number of cans is greater than 0, do everything between the curly brackets. Stop when there are no cans left.
Add the next line of the song to the variable lyrics, using the string concatenation operator "+".
End the line with a HTML line break.
Do it again—after all that's how the song goes, right?
Add the next verse, again using concatentation.
If there's still a can left (that is, the value of cans is greater than 1)...
... add the last line.
otherwise, there are no cans left...
... so add "No more cans" to the end of lyrics.
Reduce the number of cans left by 1
We've stored all the lines to the song in the variable lyrics, so now we tell the web page to write it, which just means the string is added to the page so you can see the song.

A Test Drive

You didn't think you'd do all that hard work on the exercise without giving JavaScript a try for real, did you? What you'll need to do is take the code from the previous page and type it (along with the HTML below) into a file (like index.html), and then load it in your browser. You'll see our results below:

Remember, to download all the code and sample files for this book, please visit http://wickedlysmart.com/hfhtml5.

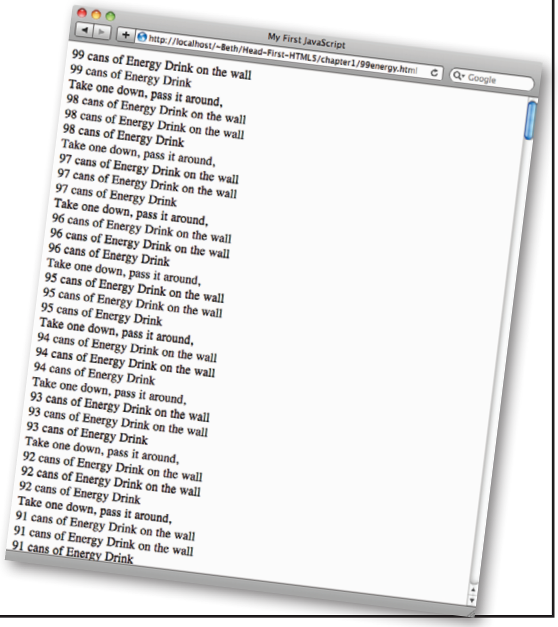
Type this in.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My First JavaScript</title>
  </head>
  <body>
    <script>

    </script>
  </body>
</html>
```

The <script> and </script> tags surround the JavaScript code. They tell the page that what's in them is JavaScript, not HTML.

And type the JavaScript code from the previous page in here.

Here's our test run of this code. The code creates the entire lyrics for the 99 ~~bottles~~ cans of ~~beer~~ energy drink on the wall and writes the text into the browser document.



My First JavaScript
http://localhost/~Beth/Head-First-HTML5/chapter1/99energy.html
Google

```
99 cans of Energy Drink on the wall
99 cans of Energy Drink
Take one down, pass it around,
98 cans of Energy Drink on the wall
98 cans of Energy Drink on the wall
98 cans of Energy Drink
Take one down, pass it around,
97 cans of Energy Drink on the wall
97 cans of Energy Drink on the wall
97 cans of Energy Drink
Take one down, pass it around,
96 cans of Energy Drink on the wall
96 cans of Energy Drink on the wall
96 cans of Energy Drink
Take one down, pass it around,
95 cans of Energy Drink on the wall
95 cans of Energy Drink on the wall
95 cans of Energy Drink
Take one down, pass it around,
94 cans of Energy Drink on the wall
94 cans of Energy Drink on the wall
94 cans of Energy Drink
Take one down, pass it around,
93 cans of Energy Drink on the wall
93 cans of Energy Drink on the wall
93 cans of Energy Drink
Take one down, pass it around,
92 cans of Energy Drink on the wall
92 cans of Energy Drink on the wall
92 cans of Energy Drink
Take one down, pass it around,
91 cans of Energy Drink on the wall
91 cans of Energy Drink on the wall
91 cans of Energy Drink
```

there are no Dumb Questions

Q: Why was there nothing in the body of that HTML except the script?

A: We chose to start with an empty body because we created all the content for this page using JavaScript code. Now, sure, we could have just typed the song lyrics directly into the body element (and that would have taken a lot of typing), or we can have code do all the hard work for us (which we did), and then just have the code insert the lyrics into the page with document.write.

Keep in mind we're just getting our feet wet here; we're going to spend a lot more time in this book seeing how we can take a page and dynamically fill in its content with code.

Q: I get that we built up the entire lyrics to the song, but what exactly did the document.write do and how did the text get in the document?

A: Well, document.write takes a string of text and inserts it into the document; in fact, it outputs the string precisely where the script tag is located. So, in this case document.write outputs the string right into the body of the page.

You're soon going to see more sophisticated ways to alter the text of a live document with JavaScript, but this example should give you a flavor of how code can dynamically change a page.

Q: You've been using the terms web page and web application; are they two different things? What makes something a web application?

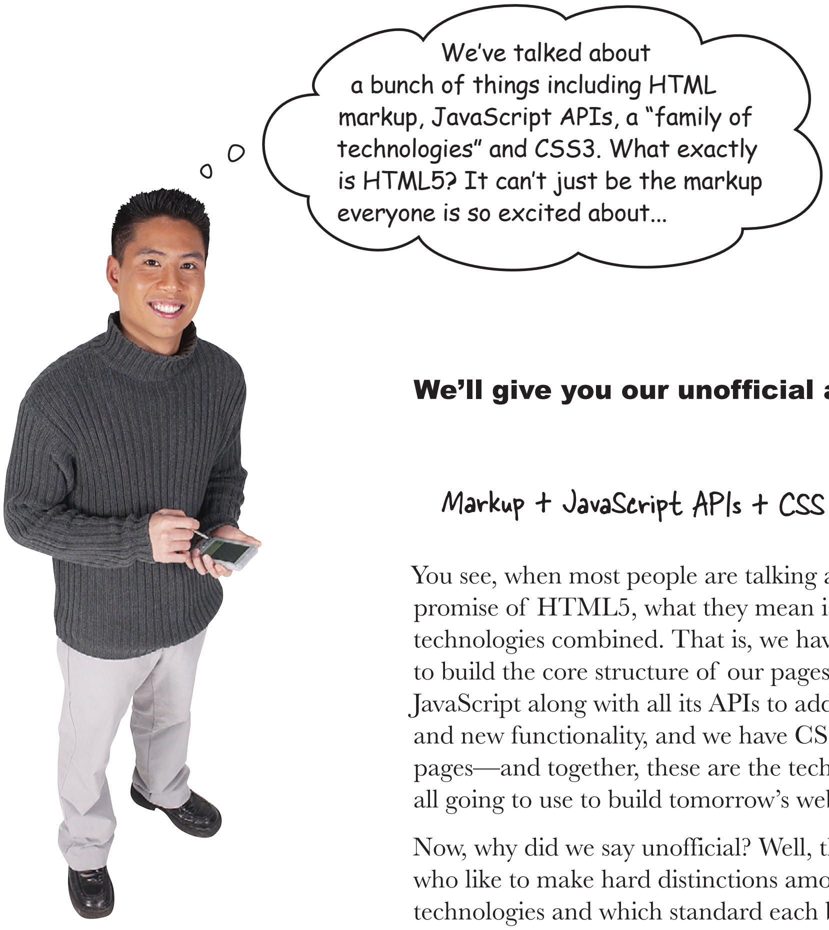
A: That's a great question because we're using the terms loosely. There's no technical difference between the two; in other words, there's nothing special you do to turn a page written with HTML, JavaScript and/or CSS into a web application. The distinction is more one of perspective.

When we have a page that is acting more like an application than just a static document, then we start thinking of it as a web application and less as a web page. We think of applications as having a number of qualities such as holding lots of state, managing more complex interactions with the user, displaying dynamic and constantly updated data without a page refresh, or even doing more complex tasks or calculations.

Q: Hey, all this JavaScript is great, but what about CSS? I'm really itching to take advantage of some of the new CSS3 stuff I've been hearing about to make my pages look better.

A: Yes, CSS has come a long way and we're thrilled it works so well with HTML5. Now, while this book isn't about CSS3, you can be sure we're going to take full advantage of some of its new capabilities. As you might know, many of the tricks we used to do to add rounded corners and shadows with images in HTML, and simple animation with JavaScript, can now be easily done with CSS3.

So, yes, we're going to make use of the power of CSS3 in this book, and we'll point out when we're doing so.


We've talked about a bunch of things including HTML markup, JavaScript APIs, a "family of technologies" and CSS3. What exactly is HTML5? It can't just be the markup everyone is so excited about...

We'll give you our unofficial answer:

Markup + JavaScript APIs + CSS = ~~Crazy Delicious~~ HTML5

You see, when most people are talking about the promise of HTML5, what they mean is all of these technologies combined. That is, we have markup to build the core structure of our pages, we have JavaScript along with all its APIs to add behavior and new functionality, and we have CSS to style our pages—and together, these are the technologies we're all going to use to build tomorrow's web apps.

Now, why did we say unofficial? Well, there are people who like to make hard distinctions among these technologies and which standard each belongs to. And that is fine and has its place. But, what we care about is this: what technologies are available in the browser, and are they ready for us to use to craft our pages and applications? So, we say HTML5 is markup + JavaScript APIs + CSS, and we think that is what people generally mean when talking about HTML5 as a technology.

If you're really interested in how these technologies fit together as a set of standards (and we all should be) then we encourage you to visit w3.org for more information.

Congratulations, you've finished Chapter 1 and written your first HTML5!

And your first JavaScript code!

Before you run off to the next chapter, we've got one more task for you to drive it all home. Use the magnets below to fill in the formula that solves the equation of "what is HTML5?" Careful now, there are some distractions thrown in with that pile of magnets. Once you've solved it, get some rest and refresh yourself before moving on to Chapter 2.

.... + ... + ... = ...

HTML5

JavaScript APIs

JavaScript

XML

CSS

Canvas

Red Vines

Markup

Mr. Pibb

XHTML

Forms

CSS3

Geolocation

<script>

Video

document

Audio

Web Workers

Offline Access

Local Storage

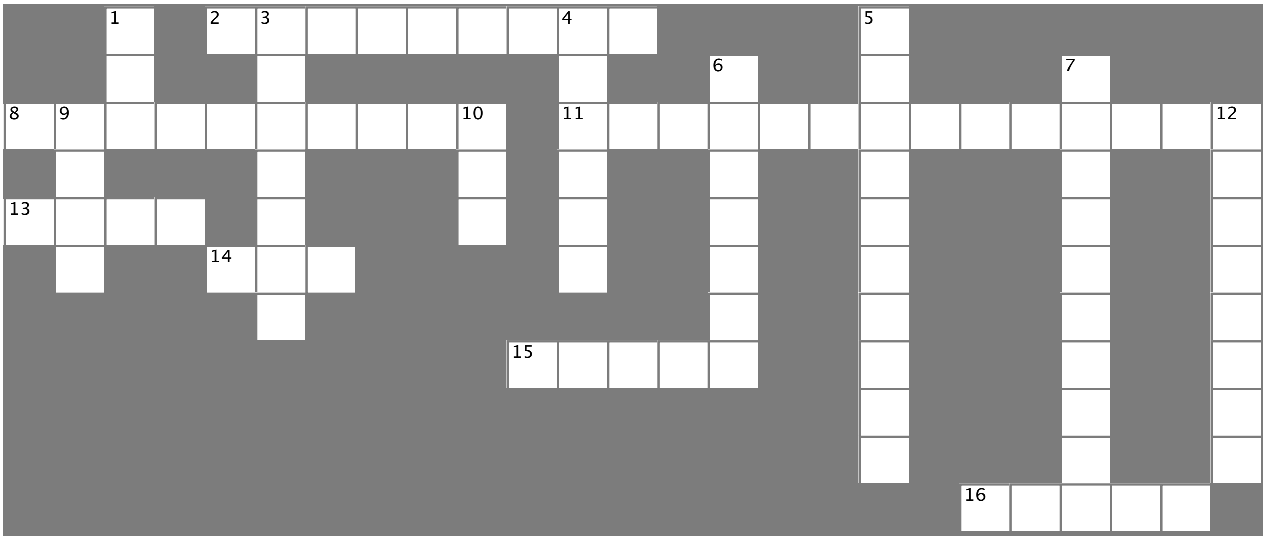
Crazy Delicious

BULLET POINTS

- HTML5 is the newest version of HTML. It introduces simplified tags, new semantic and media elements, and relies on a set of JavaScript libraries that enable web applications.
- XHTML is no longer the standard for web pages. Developers and the W3C decided to keep extending and improving HTML instead.
- The new, simpler HTML5 doctype is supported by older browsers—they use standards mode when they see this doctype.
- The type attribute is no longer needed in the <script> tag or in a stylesheet link to CSS. JavaScript and CSS are now the defaults.
- The <meta> tag used for specifying the character set has been simplified to include only the character encoding.
- UTF-8 is now the standard charset in use on the Web.
- Making changes to the doctype and <meta> tag won't break your pages in older browsers.
- HTML5's new elements are a superset of HTML 4 elements, which means older pages will continue to work in modern browsers.
- The HTML5 standard won't be officially complete until 2014, but most modern browsers will support it long before then (many support it now!).
- HTML5 introduces elements that add new semantics to your pages, giving you more options for creating web page structure than we had with HTML 4.01. We aren't covering these in this book, but we have a small guide to them in the appendix.
- Many of the new features in HTML5 require JavaScript to make the most of them.
- Using JavaScript, you can interact with the DOM—the Document Object Model.
- The DOM is the browser's internal representation of a web page. Using JavaScript, you can access elements, change elements, and add new elements to the DOM.
- A JavaScript API is an "Application Programming Interface." APIs make it possible to control all aspects of HTML5, like 2D drawing, video playback, and more.
- JavaScript is one of the most popular languages in the world. JavaScript implementations have improved dramatically in recent years.
- You'll be able to detect whether a new feature is supported in a browser and gracefully degrade the experience if not.
- CSS is the style standard for HTML5; many people include CSS when they use the term "HTML5" to describe the family of technologies used to create web applications.

HTML5cross

It's time to give the right side of your brain a break and put that left side to work. All these words are HTML-related and from this chapter.



Across

2. __________ plug could also be called spam.
8. Product that cleans up your HTML5 in three steps.
11. Your mission was browser __________.
13. The real power of HTML5 is the JavaScript ____.
14. JavaScript is ______ times faster than a decade ago.
15. Use a ______ loop to print verses of a song.
16. Got the Dear John letter.

Down

1. The ________ is an internal representation of a web page.
3. The version of HTML before HTML5.
4. The <_______> tag tells the browser what follows is JavaScript, not HTML.
5. We want our web experiences to degrade ______.
6. Much simpler than the HTML 4.01 version.
7. The standard scripting language of HTML5.
9. This attribute of the link and script tags is no longer needed in HTML5.
10. The official style standard for HTML5.
12. New ______ in HTML add semantics and structure.

WHO DOES WHAT? SOLUTION

We've already talked about the "family of technologies" so much we feel like they're, well, family. But then again we really haven't gotten to know them yet, so isn't it about time? You'll find the whole family below. Go ahead, mingle, see if you can figure out who is who. We've gone ahead and figured one out for you. And don't worry, we know this is your first time meeting the HTML5 family members, so here is the solution.

CSS3

Web Workers

Forms

Offline Web Apps

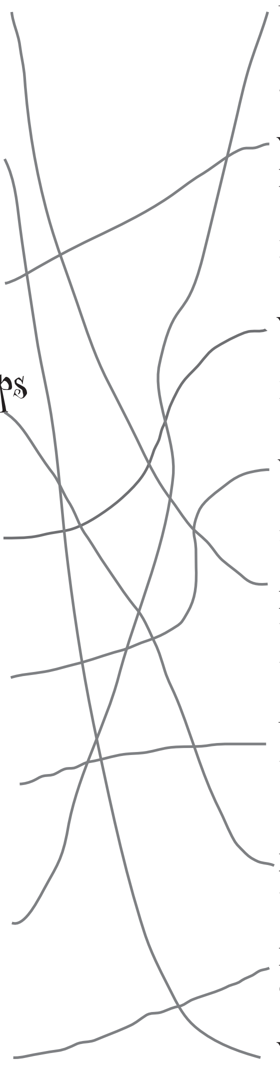
Audio & Video

New Elements

Local Storage

Canvas

Geolocation



Using me, you can draw right into your web page. With me, you can draw text, images, lines, circles, rectangles, patterns and gradients. I'll bring out your inner artist.

You might have used me in HTML 4 to enter information, but I'm even better in HTML5. I can require that you fill out fields, and I can more easily verify that you've typed an email, URL or phone number where you're supposed to.

You used to need a plug-in for us, but now we're first class members of the HTML family of elements. Wanna watch or listen to something? You need us.

We're here to help with the structure and semantic meaning of your page, including new ways of making sections, headers, footers and navigation in your pages.

I'm the most stylish one in the family. You've probably used me before, but did you know I can now animate your elements, give them great rounded corners and even drop shadows?

Use me as a bit of local storage in every user's browser. Need to store a few preferences, some shopping cart items, or maybe even stash a huge cache for efficiency? I'm your API.

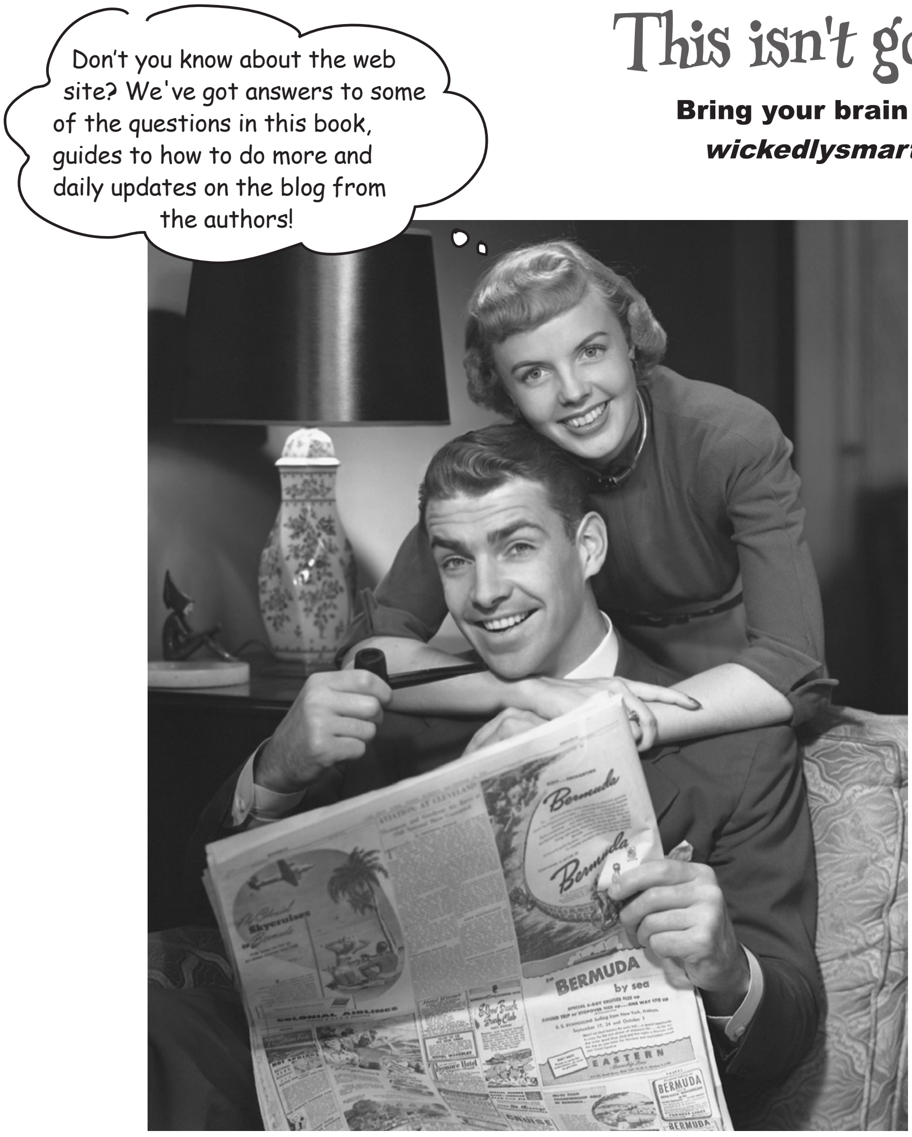
Need applications that work even when you aren't connected to the network? I can help.

I'm the API that can tell you where you are, and I play nice with Google maps.

You'll want me whenever you need several scripts running concurrently and in the background, so your user interface remains responsive.

HTML5cross Solution

		^{1}D		^{2}S	^{3}H	A	M	E	L	E	^{4}S	S					^{5}G							
		O			T						C			^{6}D			R				^{7}J			
^{8}H	^{9}T	M	L	O	M	A	T	I	^{10}C		^{11}R	E	C	O	N	N	A	I	S	S	A	N	C	^{12}E
	Y				L				S		I			C			C				V			L
^{13}A	P	I	S		4				S		P			T			E				A			E
	E			^{14}1	0	0					T			Y			F				S			M
					1									P			U				C			E
										^{15}W	H	I	L	E			L				R			N
																	L				I			T
																	Y				P			S
																			^{16}X	H	T	M	L	


Don't you know about the web site? We've got answers to some of the questions in this book, guides to how to do more and daily updates on the blog from the authors!

This isn't goodbye

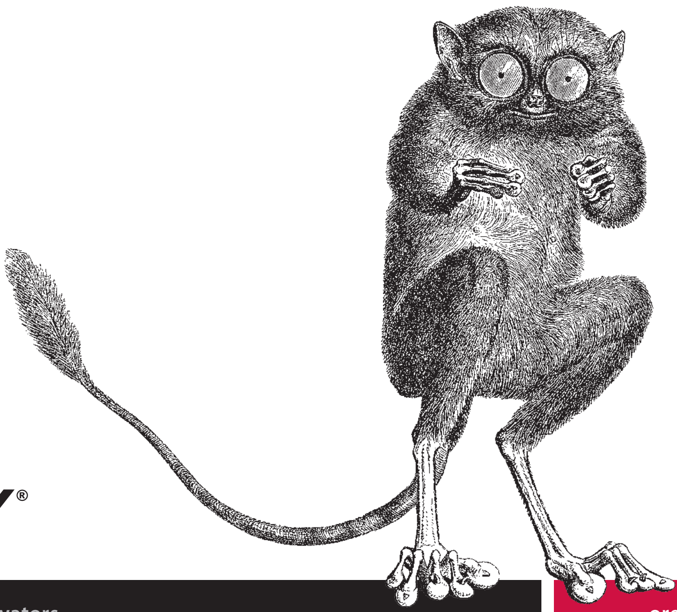
Bring your brain over to *wickedlysmart.com*

Want to read more?

You can buy this book at **oreilly.com**
in print and ebook format.

Buy 2 books, get the 3rd FREE!
Use discount code: OPC10
All orders over $29.95 qualify for **free shipping** within the US.

It's also available at your favorite book retailer,
including the iBookstore, the Android Marketplace,
and Amazon.com.



O'REILLY®

Spreading the knowledge of innovators

oreilly.com