control-shift

Jeff Wisniewski
University of Pittsburgh

HTML5

Think standards are about as exciting as tax forms? I sympathize. While you may have been hearing something about HTML5 for some time (Jason A. Clark wrote a cover story for *ONLINE* in the November/December 2010 issue, "HTML5: Changing How You Use the Web," pp. 12–14), there's a good chance you didn't drop everything, hop online, and devour the specifications as you would a good John Grisham thriller. That's understandable, maybe even healthy.

However, while previous iterations of the HTML standard might not have included enough functionality to truly exite you, HTML5 is different. It's not the Holy Grail—and it's still in flux, since it remains a draft specification—but it does exhibit some really forward-thinking and exciting changes. Webmasters should know about and, I hope, use these HTML5 capabilities. What sums it up for me is HTML5 Rocks, the title of a website (www.html5rocks.com) that says it's "created by Google" and lists 27 people from around the world, not all of whom work for Google, as authors.

There's a lot to know and learn about HTML5. Many books have been, and will continue to be, written on HTML5. In fact, Amazon lists 30 as "coming soon." Search for `HTML5` in the title if you're looking in a library collection, as the Library of Congress subject heading, `HTML (document markup language)`, is much too broad. This column will provide some background information, explain why it's important, highlight some new features, and then point you to some useful resources, both introductory and more in-depth, to help you get your toes wet.

What is HTML5? HTML5 is the newest HTML specification from the W3C (World Wide Web Consortium). Do you need to relearn HTML in order to take advantage of HTML5? Not at all. What you can code now in HTML4 will still work in HTML5. What the new specification does is put a number of new and very desirable options in your developer's toolbox, so the best way to think about it is as a "new and improved" HTML.

> "HTML5 is different. It's not the Holy Grail—and it's still in flux, since it remains a draft specification—but it does exhibit some really forward-thinking and exciting changes.

WHY HTML5 MATTERS

Conceptually, HTML5 marks a couple of noteworthy shifts. First, it puts the web browser squarely in charge, at the center of computing. It uses the browser to store information for offline use as well as to support audio and video rather than using plug-ins.

> “HTML5 storage is comparatively generous (about 5MB versus about 4K for cookies), persistent, and, of course, available offline.

Secondly, it moves to HTML tasks that were previously accomplished with bulkier JavaScripting. Previously, if you wanted to validate an email address in a web form, you probably downloaded, or wrote, a bunch of lines of JavaScript. With HTML5, this can be handled natively by a simple few lines of HTML. The HTML5 syntax is, or can be, simpler than what we've been doing with XHTML. HTML5 recognizes the importance and influence of mobile with, for example, the ability to store data locally for offline use when no data connection is available, a common occurrence in the mobile context.

Like the HTML specs that have come before it, the full HTML5 specification is pretty big—and dense. It's not exactly a compelling read. So what then are some of its standout features?

SIMPLICITY

A number of changes in HTML5 help simplify the coding process.

- The <doctype> declaration has been simplified and has gone from this:

  ```
  <!DOCTYPE html PUBLIC "-//W3C//DTD
  XHTML 1.0 Transitional//EN"
  "www.w3.org/TR/xhtml1/DTD/xhtml1
  transitional.dtd">
  ```

  to this:

  ```
  <!DOCTYPE html>
  ```

Ain't simplicity grand? Two other changes along these same lines include the following:

- You no longer need to declare a “type” for scripts and links. So instead of having to write this:

  ```
  <link rel="stylesheet" href=
  "stylesheet.css" type="text/css" />
  <script type="text/javascript"
  src="script.js"></script>
  ```

  You can now write the following (yes, I seriously like that there are a “few” less keystrokes):

  ```
  <link rel="stylesheet" href="stylesheet
  .css"/>
  <script src="script.js"></script>
  ```

- Quoting attributes is now optional, so select either this:

  ```
  <p class="mainpage">
  ```

  or this:

  ```
  <p class=mainpage>
  ```

  Either is perfectly acceptable. Personally, I'll probably continue to quote the attributes since I'm so used to seeing them that way. But in HTML5, it's optional.

STORAGE

Local offline storage isn't really new to HTML5. Technically, it's its own standard. But it is usually lumped in with HTML5, a trend I'm continuing here.

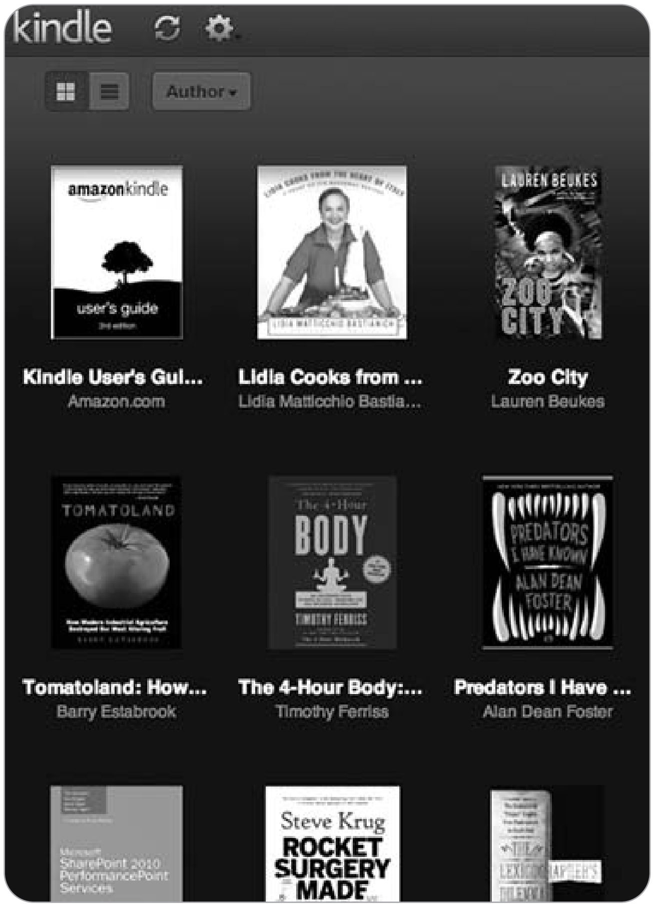
We've had local offline storage for a long time in the form of cookies. A big constraint with cookies, however, is their extremely paltry storage limit; adequate for a username and password and maybe a site preference or two, but that's about it.

Enter HTML5 storage. HTML5 storage is comparatively generous (about 5MB versus about 4K for cookies), persistent, and, of course, available offline. That's all great, you're thinking, but what's a practical use for something like this? Take a look at Amazon's Cloud Reader. Cloud Reader is a web app that leverages a lot of HTML5, including offline storage, to replicate the Kindle reading experience in the browser. One of the cool features is that, even if you go offline, your book(s) are still accessible because the text is stored in the browser (https://read.amazon.com/about).

MEDIA

One the banes of any webmaster's existence is dealing with audio and video. Has HTML5 come to the rescue? Well, not quite, but HTML5 offers the hope that browser-based audio and video playback will appear in the near future.

Currently, audio and video playback in the browser is handled by plug-ins (think QuickTime, Flash, and others), not technically handled by the browser at all. HTML5 looks to be able to change that via new `<audio>` and `<video>` tags to allow for true browser-based media playback without the need to use plug-ins. Right now, though, browser support for these tags is fragmented. Additionally, different browsers prefer different file formats (for audio, for example, Firefox wants to see an .ogg file, whereas WebKit browsers are keen on .mp3). The



Amazon's Kindle Cloud Reader HTML5 web app

specs don't support DRM, so any protected files still need to be accessed via plug-ins.

Having said all that (and that's really only the half of it), the fact that the standard allows for native audio and video support is a giant step in the right direction.

FORMS

Forms get a lot of love in HTML5. Many of the manipulations webmasters have been making to forms using JavaScript will now be able to be handled natively by the HTML. Two examples are form validation and required fields.

Up to now, if we wanted to, for example, validate email addresses in forms, we usually turned to some sort of JavaScript validation script. That's all well and good, except that even tight JavaScript validation scripts are many lines long. With HTML5, email validation can be activated by adding two words to an input field, `type=email`. Those two magic words will activate validation and have the browser check the input against standard email syntax.

Similarly, with HTML5, adding the word `"required"` in the code for an input field will prohibit the form from being submitted if said field is empty. Here's one final example—giving focus to an input field. If you want the user's cursor to automatically default to an input box on your page when it's loaded, perhaps the discovery search box on your library homepage, you can now also do that easily, without JavaScript, by simply adding the tag `"autofocus"` to the input field tag. No, you're not dreaming; this is real, and it's awesome.

SEMANTICS

The semantic web also plays a role in HTML5. Put simply, the semantic web encodes online data so that it makes sense not only to us humans but also to computers, so that a browser, for example, will know that 1234 Maple St. Pittsburgh, PA 12345 is not just text and numbers but rather an "address" and can act upon it accordingly. HTML5 takes some steps in this direction. A couple of examples would be `<figure>` and `<figurecaption>` are both used with images. The whole shebang `<IMG=example.jpg ALT="sample image">` is wrapped in the figure tag, and the caption for the image is wrapped in the figurecaption tag. *Voilà*, the browser now knows that that bit of text near the `<IMG>` is actually semantically related to it. Cool.

There's also a new tag called `<article>` that's used to designate that whatever appears with it is … drumroll … an article of some sort!

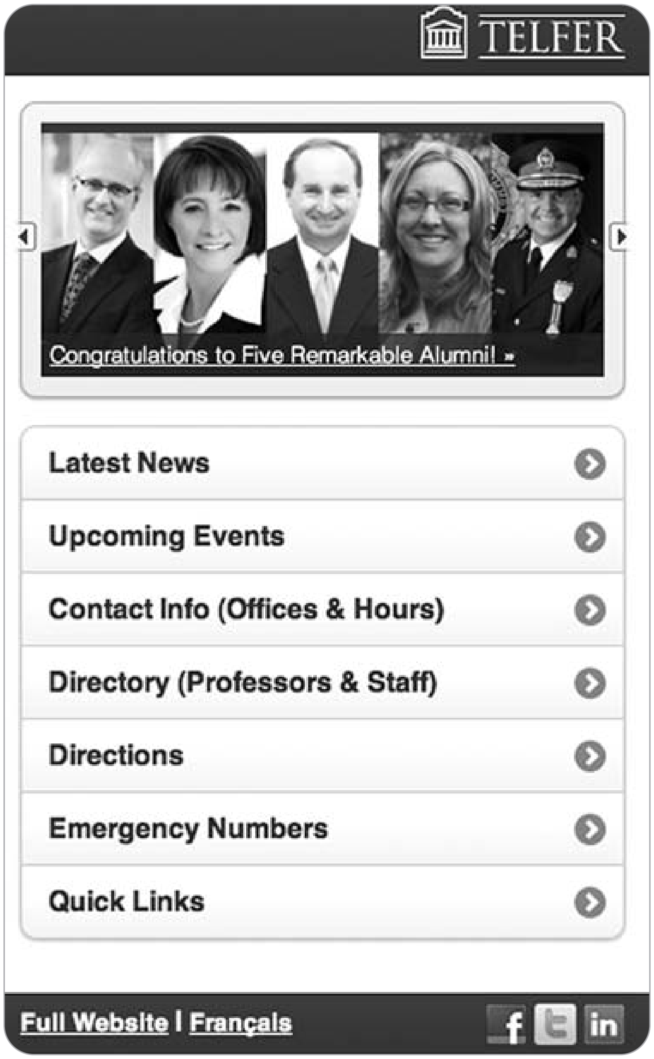
Other new semantic tags include `<header>`, `<footer>`, and `<aside>`, to name a few. While `<header>` and `<footer>` can be used for the header and footer on a single page, they can actually be used multiple times on the same page, not just once at the bottom and once at the top. For example, if your page has multiple articles on it, defined by the `<article>` tag (think blog posts), each of those can, and perhaps should, have a header tag (containing a table of contents or bylines) and footer tag (with copyright information and links to related documents, for example). As for `<aside>`, think of it as a semantically tagged callout box or sidebar. It doesn't necessarily need to be displayed as such, but that's its meaning in a nutshell.

There are other semantic tags, but these are a few examples of the way in which HTML5 tags are getting, well, more meaningful!

MOBILE

If we're talking HTML5, we need to talk mobile. Why? Because it gives developers the ability to write once and run (mostly) everywhere; it also gives them the tools to create an app-like experience in the browser. An HTML5-based web app, such as Amazon Cloud Reader, bypasses the respective app stores, app coding requirements, and app approval process of the various mobile platforms and gives the mobile user access to the "app" using just a URL. It leverages many of the tools available to developers in the HTML5 specification, most notably the offline storage feature that enables the web app to download the text of one or more of your Kindle books for offline reading.

The coding of web apps like this is not simple, but this and other mobile sites and apps like it are created using tools familiar to many library web developers: HTML, CSS,


TELFER
Congratulations to Five Remarkable Alumni! »
Latest News
Upcoming Events
Contact Info (Offices & Hours)
Directory (Professors & Staff)
Directions
Emergency Numbers
Quick Links
Full Website | Français

Telfer School of Management, University of Ottawa, mobile web app

and JavaScript. These are skills that are widely known and widely practiced, so HTML5 is to some degree democratizing mobile development.

Resources

- HTML5 Rocks by Google: www.html5rocks.com
- The HTML5 Test: How Well Does Your Browser Support HTML5?: www.html5test.com
- Swiffy from Google: Flash to HTML5 Converted: http://swiffy.googlelabs.com (Note: This is available as of this writing, but as Google is winding down Google Labs, where this project is based, its future is unclear. The site indicates it's being moved and not deleted though.)
- Imitializr: www.initializr.com (Initializr is an HTML5 templates generator to help you get started with an HTML5 project.)
- HTML5 Doctor: Helping You Implement HTML5 Today: http://html5doctor.com
- Wallaby: Experimental Flash-to-HTML5 conversion tool: http://labs.adobe.com/downloads/wallaby.html
- Safari Dev Center from Apple: http://developer.apple.com/safaridemos
- Mobile Boilerplate: http://html5boilerplate.com/mobile
- What Designers Need to Know About HTML5 and CSS3 (podcast): www.uie.com/brainsparks/2011/08/12/stephanie-and-greg-rewis-html5-and-css3
- "How to Use HTML in Your Client Work Right Now": http://html5doctor.com/how-to-use-html5-in-your-client-work-right-now

Here are the links to the standards themselves:

- HTML5: http://dev.w3.org/html5/spec/Overview.html
- Web Storage: www.w3.org/TR/webstorage

GETTING STARTED WITH HTML5 RIGHT NOW

By this point, I'm sure you're unbelievably anxious to get started with HTML5 right this very second. That's completely understandable. Where to begin? What to do? What not to do? By the time this list is in print, it may very well be out-of-date, so check the Resources section that accompanies this article for some links to sites that will have information on the most current state of browser support for HTML5. In the meantime, here are a couple of things you can get started with right away:

- Use the new doctype.
- Use the `<figure>` and `<figcaption>` tags.
- Use those form field enhancements. You may need the JavaScript as a fallback for older browsers, but these tags degrade gracefully.
- Use the semantic tags. If they're not supported in a particular browser, they'll just be ignored.

This is really just tip-of-the-iceberg stuff. I'd encourage you to explore some more by checking out the links in the Resources section. OK, maybe I was wrong when I said that HTML5 is not as exciting as a Grisham novel, because right now, I'm pretty excited about the possibilities HTML5 opens for those of us in the web community. Or maybe I'm just a big geek? Either way, HTML5 rocks.

Jeff Wisniewski *(jeffw@pitt.edu) is web services librarian, University Library System, University of Pittsburgh.*

Comments? Email the editor (marydee@xmission.com).

Copyright of Online is the property of Information Today Inc. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.