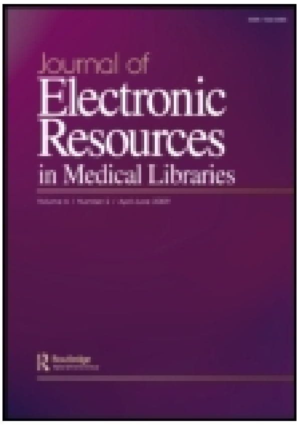

Journal of Electronic Resources in Medical Libraries

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/werm20

HTML5: Not Just a Substitute for Flash

J. Dale Prince [a]

[a] University of Maryland , Baltimore , Maryland , USA

Published online: 10 Jun 2013.

To cite this article: J. Dale Prince (2013) HTML5: Not Just a Substitute for Flash, Journal of Electronic Resources in Medical Libraries, 10:2, 108-112, DOI: 10.1080/15424065.2013.792561

To link to this article: http://dx.doi.org/10.1080/15424065.2013.792561

Journal of Electronic Resources in Medical Libraries, 10(2):108–112, 2013
Published with license by Taylor & Francis
ISSN: 1542-4065 print/1542-4073 online
DOI: 10.1080/15424065.2013.792561

eTECHNOLOGY

J. Dale Prince, Column Editor

HTML5: Not Just a Substitute for Flash

J. DALE PRINCE
University of Maryland, Baltimore, Maryland, USA

HTML5 is an under-development markup language used to structure and present web content via browser. HTML5 makes several significant breaks from HTML 4, some of which clean up syntax and some of which extend the capabilities of HTML. Recent controversies pitting HTML5 against Adobe Flash could have created the false impression that HTML5 is a Flash analog, but HTML5 is both more and less than Flash.

KEYWORDS *HTML, WWW, web, standards, multimedia*

INTRODUCTION

In 2010, Steve Jobs wrote the essay “Thoughts on Flash” and published it on the Apple Website. While it is unlikely that “Thoughts on Flash” will have the lingering fame of, for example, Joan Didion’s “In Bed” or David Foster Wallace’s “Consider the Lobster,” it caused quite a stir in the tech world in part because “Steve Jobs,” but also because it reinforced Apple’s elevation of HTML5 over Adobe’s Flash. One of the pervasive topics of discussion in the technology world in 2009 and 2010 had been Apple’s refusal to put Adobe Flash on its iOS devices. Apple stubbornly persisted in this stance through 2010 when the first iPad was released. It was at this time that Steve Jobs

Address correspondence to Column Editor J. Dale Prince, MA, MLS, AHIP, Executive Director, National Network of Libraries of Medicine Southeastern/Atlantic Region, University of Maryland, Baltimore, Health Sciences and Human Services Library, 601 W. Lombard Street, Baltimore, MD 21201. E-mail: jprin001@umaryland.edu defended Apple's decision with "Thoughts on Flash." This corporate skirmish and the marketing strategies of Apple and rival Android OS hardware manufacturers first brought HTML5 to the attention of many in the nontech world (again because, "Steve Jobs"). But it is likely that few really cared about HTML5 or Flash except when cat videos failed.

Apple's conviction notwithstanding, it is telling that Gartner's "Hype Cycle for Emerging Technologies" (for more information on the Gartner Hype Cycle, see the *eTechnology* article "Keeping Up with Technology: Sources for Mid- to Long-Term Planning," *Journal of Electronic Resources in Medical Libraries,* Vol. 9, Issue 4) did not acknowledge HTML5 in 2009 or 2010; the standard was still very much in nascent form. Indicating larger public notice, HTML5 appeared at the upward slope of Gartner's "peak of inflated expectations" in 2012. This peak is where overenthusiasm and unrealistic expectations often lead to disappointment. Regardless, it is probably a mistake to think that HTML5 is bound to be forgotten, superseded, or a victim of its own success. The language, after all, promises to become a standard for a nearly universal medium, and many existing web browsers and websites have already integrated elements even though HTML5 is only in Working Draft state. The designers of HTML5 have tried to clear away the clutter of added plugins (e.g., Flash) and extensions, increase native multimedia capabilities and interactivity, and provide support for mobile platforms. The quick uptake of these elements by web and browser developers proves that HTML5 is, at first blush, very alluring. See Figure 1.

WHAT'S NEW

Through the years, web developers have originated a number of strategies to deal with the shortcomings of HTML. Because HTML ignored multimedia, developers produced plugins for movies and interactive services such as Bejeweled, for example; these plugins included Flash, Quicktime, and RealPlayer. HTML did not realize that users might need to store more information than could fit in a "cookie," so extensions such as Google "Gears" were created to make Web apps behave like desktop apps. For each of these failures in vision, users would have to add more and more software to their browsers in order to optimize their web experience. Finally, HTML never anticipated that the web would become a media center or that it would "live" anywhere but on computers.

When the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG) began work on HTML5, they set out with a number of goals that would bring HTML in line with the way the world was actually using the web. These goals included: new features should be based on open standards, the need for external plugins should be reduced, markup should replace scripting when possible, and HTML5 should be device independent. These goals have resulted in a number

FIGURE 1 HTML5 logo. (Color figure available online.)

of differences between HTML 4 and HTML5. Among these are several that are not necessarily visible to the web user. For example, there are new structural elements that better describe the parts of a qeb page. Other elements have been eliminated due to HTML5's reliance upon Cascading Style Sheets to provide, for instance, description of text. Features that are more readily evident to web page users are tags that allow easy inclusion of audio, video, and interactive elements and do not require the user to install an additional plugin to make the fun parts of the qeb work.

Simplified Syntax

To get the boring stuff out of the way, HTML5 is offering quite a few things under the hood that will not be visible to the layperson's eye. First, HTML5 has simplified its syntax over HTML 4. No, it is not as simple as it was back in the late 1990s before things such as DOCTYPE declarations were a requirement, but even that type of declaration is much simpler. There are three ways of declaring DOCTYPE in HTML 4: 1) HTML 4.01 Strict, 2) Transitional, and 3) Frameset (with an additional set of around four declarations for XHTML). The HTML 4.01 Transitional DOCTYPE declaration looks like:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN”
“http://www.w3.org/TR/html4/loose.dtd”>
```

Note that the declaration contains a URL (http://www.w3.org/TR/html4/loose.dtd), which refers to a DTD or a set of rules for the markup language.

This was included because languages based on SGML require a DTD. HTML5 is not based upon SGML (unlike the HTMLs up to 4), so it does not require a DTD. Thus the DOCTYPE declaration for HTML5 looks like:

```
<!DOCTYPE html>.
```

One could well ask why, if all that other detritus can be cleared away, there needs to be a DOCTYPE declaration at all? However, the declaration lets a web browser know that it is looking at a web page and should behave following the standardized rules of reading a web page.

So, if the reader has not given up, eyes glazed over upon seeing HTML show up in the middle of the article, HTML5 has also simplified character encoding, script, style, and link elements and has reduced the requirement to use the div element by adding some new semantic components that make layout simpler: header, footer, article, and section. These elements, fairly self-explanatory, will allow search engines to delineate content and its weighted value more easily. In summation, it is easier to write HTML5 and it makes creating discrete page parts easier.

Offline Functionality

Cookies have been the traditional way for browsers to store information about users, but cookies—unenhanced cookies, that is—are very small in size and are meant to be read by the server. They contain identifying information or information such as the last time a site was visited—all information that needs to be processed by the server. The inability to store information on the user's computer makes it impossible to, for instance, work offline with a website. Two new technologies have been added by HTML5 to make temporary offline work more graceful than historically so: application caching and client-side, or offline, storage. Application caching is much like an existing browser cache, but is intended not to be overwritten with ease. This type of storage allows nonuser-specific information to be stored in a special cache. So the active parts of a text-editing program can be stored in the application cache, making it possible to make use of the editor while offline. Client-sided storage allows users to work offline and sync with the server later. This kind of storage can also speed up performance if an initial set of data is downloaded once and does not have to be copied again. Drawbacks include the fact that offline data cannot be accessed by other clients, so a document edited on an offline browser at work cannot be accessed from a smartphone unless the document has been synced online.

Audio/Video/Multimedia

Given the contretemps between Apple and Adobe (iOS v. Android/HTML5 v. Flash), it would be unsurprising if mainstream users thought that HTML5 was equivalent to Flash, but the two have little in common. Flash is a multimedia platform that allows for the creation of animated or interactive objects while HTML5 is, as HTML has always been, a markup language for structuring and presenting content. Only HTML5 makes inclusion of interactivity easier than previous iterations of HTML and provides a tool—the Media API—that facilitates this inclusion. The argument stemmed from the fact that Flash works with a browser plugin, while HTML5, typical of the goals of the HTML5 working group, provides the ability for an enabled browser to play, for example, audio or video without plugins, using just the APIs and JavaScript to perform a large number of multimedia actions. The two elements in particular are the <video> and <audio> elements. Provided the browser has been built with HTML5 in mind, embedding a movie or audio file and making use of the browser's built-in player is simplicity itself. However, there are battling standards for those video and audio formats, and while the HTML5 working group has made recommendations for formats that are patent and royalty-free, some browser developers have interest in supporting proprietary formats. At this writing, Apple's Safari browser, which claims to be HTML5-compliant, does not support the OGG container, an audio format recommended by the working group.

HTML5 also debuted the <canvas> element, an element that is used to draw 2D graphics on the fly. Also true to HTML's history as a markup language, <canvas> creates only a container for graphics without providing any of the actual ability to draw. In order to take advantage of this space, a developer will have to use, for example, JavaScript to actually draw the graphics. Nevertheless, very interesting effects can be created with this element like interactive graphics that respond to cursor moves.

CONCLUSION

HTML5 provides many other tools, especially in the form of APIs, for web developers to create modern web apps that act as if they were desktop applications, for instance, the Drag-and-Drop API that lets a developer call upon routines that will allow users to pull content off or drop content onto a website with the click of a mouse. Another highly discussed API is the Geolocation API that permits a website to know where a user's location. This API has caused considerable excitement in mobile-app developer circles and could have a section all to itself in this article; however, not to be pedantic, the Geolocation API is not part of the HTML5 specification and so was not covered here in depth. Nevertheless, this API will be used in tandem to other HTML5 elements to create many new and exciting mobile applications in the near future. Currently, and not surprisingly since HTML5 is not a mature product, adoption is piecemeal, and it is likely that this piecemeal adoption will continue for some time. This means that great things will happen on the web, and users will be stymied because they cannot play the OGG-formatted audio from Wikipedia in their favorite browser even though it claims to be HTML5 compliant.