



HTML5 is for Web Design, not Apps

HTML5 is the first new version of HTML to come along since 4.0 was introduced in 1997. HTML (Hyper Text Markup Language) and HTML elements are the basic building blocks of web pages. Without HTML, the Internet as we know it would not exist.

However, HTML is not a programming language at all, a confusing concept to many. Web pages rely on a scripting language, such as JavaScript, in order to respond to user input. Looking for a visual? Think of HTML as a car and JavaScript as the engine.

Not coincidentally, JavaScript was introduced in 1995, the same time the web really started to "take off." This is an important point to consider before we dig deeper into HTML5.

The open source nature of HTML and JavaScript has contributed to the growth of the Internet as a whole, as source code and design ideas are often "borrowed" from around the web and even from competitors, as they are impossible to hide and protect. You can obfuscate JavaScript, but you simply cannot prevent a competitor from reusing your code on another website. That is a fact.

There has been a lot of industry talk about HTML5's place among the top candidates for financial app writing. But as with most new technologies, the discussion has been littered with confusion and misinformation. So, when should HTML5 be used? And for what purposes?

New Features – Same Old Concept

HTML5 includes new features such as the "Canvas" element, along with audio, video and Scalable Vector Graphics (SVG) content, some of which replace the "object" tags of HTML4. There have also been some modifications and standardization of tags, such as "a," "cite," "menu," etc. However, the basic concept remains the same.

Who's Promoting HTML5?

HTML5 was originally proposed by Opera Software, the producer of the Opera web browser, and designed by the World Wide Web Consortium (W3C), an organization criticized as being dominated by large backers with standards to suit.

Those large backers, including Apple, Microsoft, Adobe, Google, and Facebook, have been some of HTML5's greatest cheerleaders. In fact, the late Steve Jobs termed HTML5 as the win-win solution for consuming any kind of web content on all types of web platforms. It is also important to note Apple, Microsoft, Google and Facebook each have an incentive for promoting HTML5, owning and operating very profitable "app" stores such as Windows Marketplace, Google App Store, iTunes App Store and Facebook Canvas Apps. Protection of intellectual property and trade secrets belonging to app developers is of little or no concern to these magnates; in fact, it is a conflict of interest.

Fact #1: HTML5 is Based on JavaScript, which is Impossible to Protect

As previously discussed, JavaScript is the core "language" of HTML5, a source code which is *impossible* to protect. The most evident reason: anyone can view HTML5 and JavaScript source by just a simple mouse click. JavaScript was never intended to be anything more than a scripting language for web pages and an engine for client-server communication and re-useable web content, including menus, buttons, tabs and the like. It was *never* designed for business logic.

Developers may argue they can mangle JavaScript source code by obfuscation and pseudo-encryption (such as minification, where the code is compressed and "minified"). This is a dangerously false sense of security. Any skilled developer can reverse engineer a "minified" or "obfuscated" JavaScript application. By design, JavaScript can be neither encrypted nor hidden. JavaScript files must be downloaded in order to run in a web browser.

To make matters worse, JavaScript cannot be locked down to a specific web domain. Developers can make attempts to lock code, but those efforts can be very easily circumvented.

The code is accessible to be hacked by any competitor. No matter how many steps are taken to hide the JavaScript, the web browser or OS eventually downloads the JavaScript file to run it within the client's web browser or OS.

For comparison, hacking a Windows EXE or Objective C iOS app can be difficult if the developer has taken proper precautions, but HTML5 apps can be stolen in minutes by children.

The hacks of famous HTML5 games like Angry Birds, Texas Holdem and many others are excellent examples of the insecure techniques employed by HTML5 developers. All security measures are rendered useless once the code has been obtained and reverse engineered. No matter what any HTML5 or JavaScript developer may claim, these facts are true and irrefutable!

Developers also argue business logic should reside on the server, because with HTML5, there is no other option. If business logic resides on the server, then a financial app cannot be considered "real-time", as it demands substantial network transmission delays.

Further, placing business logic and sensitive information on cloud-hosted servers can be disastrous, especially for financial applications. For example, on September 2, 2014, actress Jennifer Lawrence confirmed a leaked topless photo was stolen after her Apple iCloud account had been hacked. Apple confirmed cloud hosting accounts were compromised, including those of other celebrities, as well.

Fact #2: The Concept of HTML5 Canvas is Nothing New

HTML5 is based on the same concepts as HTML4. The most useful and talked about feature of HTML5 is the "Canvas" object. Canvas makes it easy for JavaScript developers to "paint" within a web browser, simplifying the design of applications that utilize graphics. You may be surprised to know that this functionality has been available in a simulated manner, with little attention, since the release of HTML4 in 1997.

Fact #3: There is a Widespread Misconception about HTML5

HTML5 has been labeled as a "cross-platform programming language," a gross misconception. It is true: HTML5 is capable of running on all types of platforms, such as Windows, Linux, Android and other mobile operating systems. But, let's not forget HTML5 is nothing more than web page markup, controlled by JavaScript.

Classifying HTML5 and JavaScript as a true programming language would be totally unfitting. It is a browser integrated language which cannot utilize all the features of the underlying operating systems. This may change with future versions of operating systems, but the security implications would ensure that apps would be quite annoying to users (recall Windows Vista security alerts), and the performance of JavaScript will never be equal to native languages.

When Should You Use HTML5?

Use HTML5 when you do not care about source code, intellectual property, or trade secrets. Use it for webpage design or for a free app which doesn't require security. HTML5 is great for buttons, menus, and website content, which are its intended uses.

The Bottom Line - Better Options Available

The bottom line is that HTML5 and JavaScript are great for web design, but trading apps built on these technologies are bound to be copied and exploited by other parties, as these technologies were not designed for developing business applications.

HTML5 is fine if all you need is a simple chart to display on your website.

If you're creating a mobile trading application, there are far better options for iOS and Android development, which offer better protection of trade secrets, better security, better performance, increased usability, and also allow usage of the native features of the underlying operating systems. For iOS (iPad and iPhone), use Objective C, and for Android, use Java.

For desktop development, consider C++, C#, Delphi or other robust languages.

Objective C, C++, C#, Java and other low-level languages are always the best option for desktop and mobile applications. JavaScript is best suited for web design.

Permission to Redistribute

Copyright © 2012-2014 by Modulus <http://www.modulusfe.com> - All rights reserved. Permission is hereby granted to redistribute the article providing this copyright notice remains in place.

About the Author

Richard Gardner is Founder and CEO of Modulus Financial Engineering, an influential member of the financial technology industry, and a world-renowned professional trader and software engineer. Connect with Richard on [LinkedIn](#).