

A CAPLIN WHITE PAPER

# 2012: The Year of HTML5

BY PAUL CAPLIN



**CAPLIN**

[www.caplin.com](http://www.caplin.com)

FEBRUARY 2012

# One of the most striking things about the coming year is its unpredictability.

What's going to happen to the Eurozone?

Who is going to be in the White House?

Will the global economy recover?

What will happen in Iran?

There is uncertainty everywhere you look. But in the world of software at least, one thing looks like a safe bet: in 2012, HTML5 will emerge as the world's favourite GUI technology and, in particular, the tool of choice for building Web trading applications. This white paper explains why this is going to happen, and why it's a good thing.

## BACK TO THE FUTURE

The Web first took off in the early 1990s as a way of publishing text, and was soon extended to include images. Later scripting was added, so that Web pages could be interactive, and then multimedia content. With each additional capability, the Web became capable of handling entire new categories of activity – and in each case, it came to dominate that category, driving out installed applications.

Why? Because Web-delivered content is more widely accessible, compatible with almost all devices, easier to update, easier to find via search engines, more convenient for users, cheaper to deliver, easier to integrate with other content... the list goes on

and on, which is the reason that traditional applications don't stand a chance in areas where the Web works well.

Two areas in which the Web did not work well in the past were graphical animation and rich interactive applications. These were pretty obvious shortcomings, and many firms tried to fill the gap using plug-ins and bolt-ons. Some of the more successful efforts included (in rough chronological order) Java applets (from Sun), ActiveX (from Microsoft), Flash (from Macromedia/Adobe), Flash's big brother Flex, and Silverlight (from Microsoft again). Each of these was an attempt to provide a workaround for one or both of the above



Paul Caplin, CEO of Caplin Systems Ltd. founded the company in 2000

failings – or, more cynically, to get a proprietary lock on a key area of Web functionality. Each was successful for a time.

But they all suffered from being closed, private technologies that didn't integrate at all nicely with the rest of the Web. So when, in 2007, the plan for HTML5 was announced – an upgrade to the way the web worked that would decisively plug these gaps within the browser, in an open, Web-friendly way – it was pretty clear that this would, eventually, replace the zoo of proprietary plug-ins. Apart from a few diehard disciples of Adobe or Microsoft, almost no one disputed this

The big question was: when?

## COMMUNICATION ERROR

Back in 2008, a statement from Ian Hickson, an official maintainer of the HTML5 specification, caused an awful lot of confusion, some of which endures to this day. Talking about the expected HTML5 timeline, he said<sup>1</sup> that there would be a “candidate recommendation” in 2012, a “final version of the test suite” in 2019, and a “proposed recommendation” in 2022. The plug-in vendors seized on this as meaning that HTML5 was still a decade and a half away from being a reality, and this view was widely promulgated.

But this is a misunderstanding of what the words “proposed” and “candidate” actually mean in the rarefied world of

technical standards. The 2022 date refers to the finalisation of a huge and incredibly detailed public testing suite, which is of largely specialist interest; the key date, on which HTML5 was forecast to be ready for commercial use, was 2012.

But even that date turned out to be wrong, because in the event the major browser manufacturers (Apple, Google, Microsoft, Mozilla) didn’t wait for the standards bodies to finish their debate. Instead, they raced ahead and implemented HTML5 anyway, thrashing out the details among themselves. In a few areas this produced incompatibilities between browsers, which are now being ironed out; but to a surprising

extent, it resulted in a high-quality, consistent de facto standard rapidly emerging.

And to cope with the inevitable discrepancies, powerful libraries like jQuery and tools like H5BP emerged, providing an abstraction that works in any browser and neatly hiding any differences in implementation.

In effect, there was such a large commercial benefit to every software firm in converging on a spec as fast as possible that the market didn’t wait for the official pronouncement: it simply solved the problem for itself. The standards bodies were then left to document what the market had already decided.

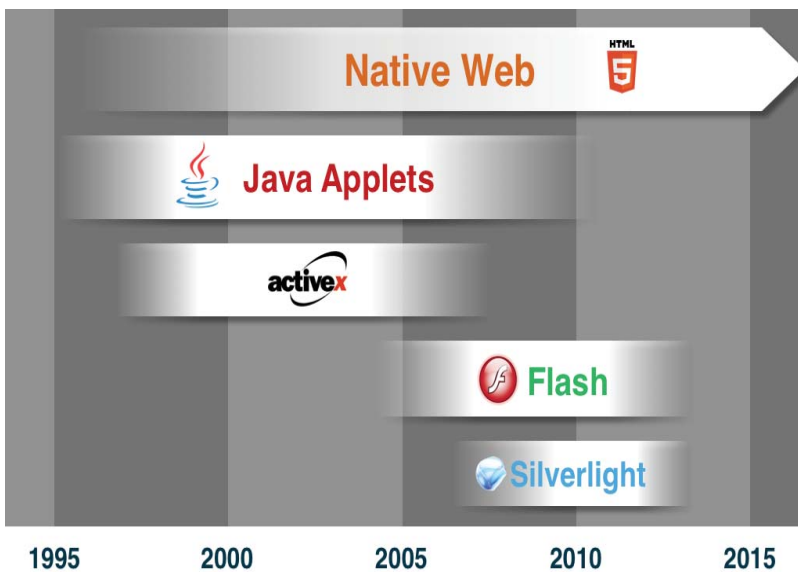


Fig. 1 – Technology lifecycles for Web-delivered apps

“The world is moving to HTML5”  
Steve Jobs, CEO, Apple

## IT’S LATER THAN YOU THINK

From 2008 onwards, the race to produce a new generation of HTML5 browsers was on, and no software company wanted to be left behind.

Starting that year, Google backed<sup>2</sup> HTML5 to the hilt in almost every area. In early 2010, Steve Jobs effectively bet his company on it, when he banned Flash on all new Apple devices and declared that HTML5 was the future.<sup>3</sup>

At first, Microsoft dug its heels in, releasing IE8 as a half-hearted stab at an

HTML5-compliant offering, and heavily promoting its Silverlight plug-in as a competitor to Flash. By the second half of 2010, though, even Microsoft could see the writing on the wall.<sup>4</sup> It started backpedalling furiously on Silverlight,<sup>5</sup> and simultaneously raced ahead with IE9, its first truly competitive browser for ten years. It even launched its own campaign to rid the world of IE6.<sup>6</sup>



Microsoft bakes a cake to celebrate IE6 market share falling below 1% in the USA

<b>June 2004</b>	The Web Hypertext Application Technology Working Group (WHATWG) is formed to develop a new standard for HTML, to be called HTML5.
<b>May 2007</b>	The World Wide Web Consortium (W3C) abandons its own efforts and agrees to adopt HTML5 as the next major Web standard.
<b>April 2009</b>	Google VP of Engineering Vic Gundotra announces <sup>16</sup> that Google is “betting big on HTML5,” and predicts that HTML5 apps will soon overtake Windows applications in quality of user experience.
<b>January 2010</b>	YouTube begins trialling <sup>17</sup> HTML5 as a replacement for Flash.
<b>April 2010</b>	Steve Jobs publishes his famous open letter <sup>18</sup> explaining why Apple will never support Flash (or Silverlight) on its new devices, and says “Apple has adopted HTML5.” The war on plug-ins begins.
<b>October 2010</b>	Microsoft CEO Steve Ballmer publicly states <sup>19</sup> that “the world is going HTML5... and so are we.” Soon afterwards, Microsoft starts to phase out Silverlight.
<b>March 2011</b>	Microsoft dedicates itself <sup>20</sup> to eradicating the use of IE6, the 10-year-old pre-HTML5 browser that has become a millstone around its neck, and launches the ie6countdown <sup>21</sup> website.
<b>June 2011</b>	Microsoft reveals <sup>22</sup> that Windows 8 will use HTML5 as the main app development platform for mobile, and will also offer it for the desktop.
<b>August 2011</b>	The Financial Times becomes one of the first major content providers to stake its future on HTML5, when it replaces <sup>23</sup> its iPad app with a Web app. It is a huge success, hitting 1m users <sup>24</sup> within three months. Online sales account for around 25% of the FT’s income.
<b>August 2011</b>	Microsoft steps up its marketing of what it calls plug-in free browsing <sup>25</sup> in the forthcoming Windows 8 Metro – no Silverlight, no Flash, just HTML5.
<b>August 2011</b>	Google announces <sup>26</sup> that Google Apps will no longer work in browsers that don’t support HTML5.
<b>September 2011</b>	Microsoft announces WinRT for HTML5 development on Windows, and describes <sup>27</sup> HTML5 as “the modern style of Windows development.”
<b>November 2011</b>	Adobe states <sup>28</sup> that it is abandoning Flash for all mobile devices and will now “aggressively contribute to HTML5” as “the best solution for creating and deploying content in the browser across mobile platforms.”
<b>December 2011</b>	Microsoft has a party <sup>29</sup> to celebrate IE6 share falling below 1% in the US, while worldwide market share <sup>30</sup> for IE6 falls to 1.8% and for IE7 to 4.0%.
<b>December 2011</b>	Morgan Stanley reveals <sup>31</sup> that it is trialling Google Chrome Frame as a way of enabling its legacy browsers to handle HTML5.
<b>January 2012</b>	Spanish banking group BBVA announces <sup>32</sup> that is migrating its entire 110,000-strong workforce from Microsoft Office to HTML5-based Google Apps.
<b>January 2012</b>	Chrome overtakes <sup>33</sup> Firefox to become the world’s second favourite browser, and looks set to replace Internet Explorer as the favourite within the next few months.

Table 1 – The timeline above shows some of the key milestones in the advance of HTML5, and the retreat of the plug-ins.

## REASONS TO BE CHEERFUL

So... what exactly is HTML5? What is all the fuss about? Why is it attracting such a huge level of investment worldwide?

The first point to make is that when people use the term HTML5, they aren't usually talking about just HTML5. The word is widely understood as shorthand for an integrated bundle of technologies that includes not only HTML5 itself but also CSS3, Canvas, ECMAScript 5, WebSockets and several others; and beyond that, to stand for the whole concept of running native apps in a browser.

The actual HTML5 part makes complex Web pages much easier to create and maintain, and allows them to look and perform just like applications. The CSS3 part makes it easy to style the result in any way you like and to add complex visual treatments and transitions. The Canvas part is for creating complex graphics, still or animated.

ECMAScript 5 is a better version of the JavaScript language that Web developers have been using for years.

Although standardisation work continues in a number of areas, this bundle of technologies is already supported in an impressively uniform way across virtually all modern browsers, and where discrepancies exist there are plenty of handy abstraction tools that mean you don't have to worry about it. The result is that developers can now build applications – proper applications, that do anything you want and run quickly and smoothly – right in the browser. HTML5 apps load like Web pages, and just work. You can use them on your PC, your Mac, your HTC, your iPad, your BlackBerry Playbook, your Xbox, your new TV – anything that can access the Web. In a world where most computers are now not PCs, that is very important.

For the user, that means they can use the app whenever they like, anywhere, on anything. They don't have to install or maintain it, and they automatically get the very latest version every time they use it. For the provider, it means they only have to write the app once, and it will run everywhere, without having to be installed on the user's machine with all the support headaches that entails. It's the Holy Grail of application development, at last.

And it works. At Caplin we have been building high-end trading applications in HTML5 (and its recent predecessors) for the last few years, and they are in everyday production use at major banks with tens of thousands of end users.

“If you want to do something universal, there is no question, the world is going HTML5. That is clear... the world is just pushing down this HTML5 path, and so are we”

Steve Ballmer, CEO, Microsoft

## WHAT'S THE CATCH?

Is HTML5 over-hyped? Maybe it's not really ready yet. Or maybe it creates as many problems as it solves.

Well, the fact is, it really does work. There are hundreds of cool HTML5 demos and websites out there, but in terms of serious business applications you only have to look at Google Apps to see that HTML5 is absolutely up to the job, and is increasingly used by large corporates around the world (BBVA is currently rolling out Google Apps<sup>7</sup> to 110,000 staff as a replacement for Microsoft Office).

And many leading banks such as Citigroup, Barclays Capital and Deutsche

have successfully deployed native browser apps for mission-critical real-time trading applications.

So, why isn't everyone using it? Apart from natural inertia, there are three reasons commonly cited for hanging back. They are:

- There are still older browsers in use in the real world that don't support it.
- HTML5 is still a work in progress, not yet ready for the mainstream.
- Immaturity of development tools and lack of suitable skills means that developers are not as productive in HTML5 as in older technologies.

All these arguments once had validity. To understand why the HTML5 dam is about to burst, let's look at why none of them stands up any longer.

**“PEOPLE DON’T HAVE HTML5 COMPLIANT BROWSERS!”**

Don’t they? The numbers show that, worldwide, the proportion of browsers that adequately support HTML5 has risen over the last 12 months from 53% to 72% (see Figure 2 below). And a full 94% of the browsers in general use now support at least some of HTML5, if not all of it.

In the wider market, HTML5 is now standard on all smartphones, tablets and Macs. It’s available on all PCs – Google Chrome runs on pretty much everything, is free, is exceptionally secure,<sup>8</sup> and only takes a few minutes to download and install. The only obstacle to universal use is now inertia, mainly on the part of late-adopter consumers and large firms.

Of course, overall adoption rates for large corporate users are lower. For a few very large firms it remains hard to upgrade from older Microsoft browsers, because of

organizational inertia and because of the need to remain compatible with antiquated internal websites that might not work in modern browsers. You may be concerned that your particular user base happens to contain more than its fair share of the 1% who are still stuck on IE6.

There are two reasons why this problem will go away in 2012.

The first is a plug-in for Internet Explorer called Google Chrome Frame. This is easy to download and install, without the help of an administrator, in much the same way as the Flash plug-in. What it does is to provide all the functionality of Google Chrome – widely regarded as the most advanced and secure browser on the market<sup>9</sup> – within Internet Explorer. Once installed, it is invisible to the user, and remains dormant until needed. Most websites will continue

to function exactly as before. But when a modern web page that requires HTML5 is accessed, Chrome Frame (optionally) leaps into action and displays the page exactly as it would appear in a Chrome browser, but inside Internet Explorer.

This approach provides the best of both worlds for firms that for one reason or another aren’t yet ready to make the jump to modern browsers. It is difficult to see any rational objection to it, which is why, for example, Citigroup is now mandating it for some of its client apps and Morgan Stanley is trialing it internally.<sup>10</sup>

The second reason is that the use of HTML5 is spreading so rapidly that moving to modern browsers is very soon going to be unavoidable for all firms. For years, Web page designers have gone to considerable lengths to make sure that their designs

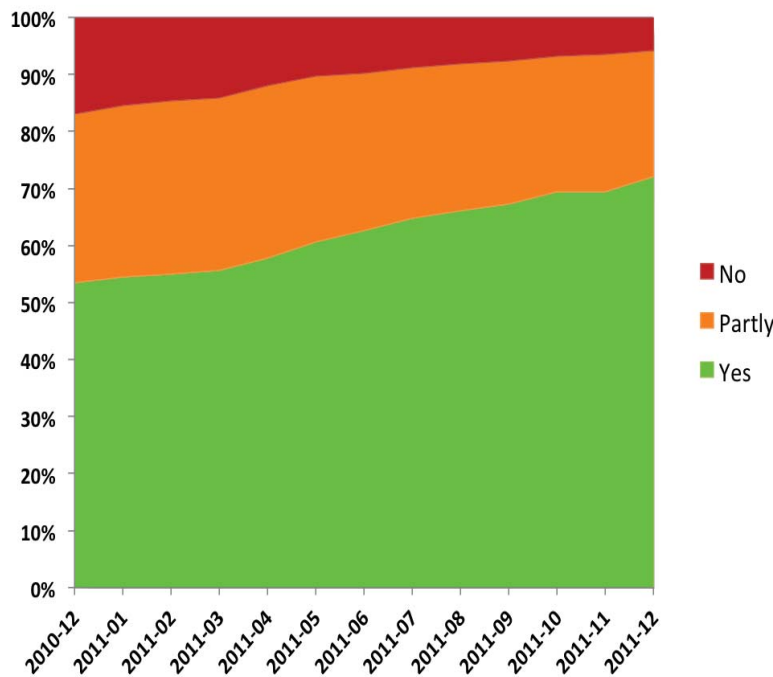


Fig. 2 – The proportion of browsers in worldwide use that support HTML5<sup>34</sup>

would work even in very old browsers such as IE6. Even though better browsers were available, internal systems managers could get away with sticking with old ones since, while they might be slow and clunky, they did at least allow users to view virtually all Web content.

But this is simply no longer the case. Web developers are starting to assume that the minority of users who are not yet

HTML5-ready very soon will be, and are increasingly reluctant to go through all the pain required to get their sites to work in old browsers. Google has already officially withdrawn support<sup>11</sup> for its Web apps in any browser more than two versions old, and many others are following suit. As a result, internal systems departments at large firms will soon have to provide newer browsers as a business-critical requirement.

	IE	Firefox	Chrome	Safari
3 versions back	6.0: 7%	6.0: 91%	13.0: 97%	3.2: 81%
2 versions back	7.0: 25%	7.0: 95%	14.0: 97%	4.0: 82%
Previous version	8.0: 44%	8.0: 95%	15.0: 98%	5.0: 90%
<b>Current</b>	<b>9.0: 91%</b>	<b>9.0: 95%</b>	<b>16.0: 99%</b>	<b>5.1: 92%</b>

Fig. 3 – Browser support for HTML5, CCSS3 and JS API standards that are at “recommendation”, “proposed” or “candidate” stage<sup>35</sup>

### “HTML5 ISN’T READY YET!”

It is sometimes said (mainly by developers with strong skills in other areas) that HTML5 is good but it’s just not ready yet. By this, they mean that there are parts of the emerging spec that aren’t supported by all browsers. This is used as a justification for delaying adoption.

This point of view may have had some validity a year or two ago, but it no longer holds water.

The above table shows the proportion of the HTML5 standard that is already implemented in the main browsers.<sup>12</sup> Excluding features that are still at “candidate” stage, all current browsers already support 100% of HTML5.<sup>13</sup>

For a great graphical view of what is and is not supported across all the browsers, and how this has evolved over the last few years, see [html5readiness.com](http://html5readiness.com)

(unsurprisingly, you will need a modern browser to view this). It can be seen that Chrome already supports virtually the entire standard, Firefox and Safari support almost all of it, and Microsoft... well, Microsoft is trying to catch up. Either it will quickly get back into the race, or people will stop using Microsoft browsers. Steve Ballmer’s aggressive comments suggest that it will be the former.

So in fact, almost all of HTML5 is now in place, except in Internet Explorer, which is catching up fast. And as pointed out in the previous section, there is a simple workaround for IE, which is to use Chrome Frame – a plug-in that is no harder to install than Flash, and easier than Silverlight.

Of course, more features will be added over time. But these features are, for the most part, completely independent of one

another, and it is not necessary for every last one of them to be in place for HTML5 to be immensely useful. HTML5 is an evolving technology, and is already very powerful. Just because it keeps getting better is not a reason not to jump on board.

Furthermore, there are now third-party libraries that fill the remaining gaps in a nice cross-browser style. The Caplin real-time framework, for example, enables low-latency real-time streaming to work well in all browsers, even the ones that don’t support Web Workers and Web Sockets.<sup>14</sup>

Finally, add to the mix the fact that most browsers are now self-updating (so that people automatically have the latest version all the time) and it becomes clear that “not ready” is fast becoming almost meaningless as an objection.

## “DEVELOPER PRODUCTIVITY IS LOW IN HTML5!”

Is it hard to build enterprise-grade apps using HTML5?

This question is not really about the nature of HTML5, CSS3 or Canvas. They are all based on modern, well-designed languages, they are all highly effective, and they all stack up well against alternative approaches.

It's not really about JavaScript either, which is increasingly popular and increasingly being treated as a first-class language. There are already plenty of high-quality enterprise applications written in JavaScript.

The determining factor in the ease of building HTML5 apps is, in fact, all about the tools, frameworks and skills that are available.

As you would expect, there has been explosive growth in this area over the last few years, with a stampede of firms large and small vying to offer the best toolkits, as well as many high-quality open-source initiatives.

Table 2 below shows a (far from exhaustive) list of some of the great tools that now exist.

When it comes to mobile, the advantage of HTML5 is already clear-cut. Forrester Research recently issued its second annual report<sup>15</sup> on the relative productivity and performance of native vs. HTML5 apps for mobile devices. Talking about the benefits of HTML5, it said:

As firms move from two to three platforms and even contemplate adding Windows Phone 7, they will find that native development becomes increasingly untenable from an economic point of view. Add in multiple releases a year as the mobile platform vendors update their operating systems, and most shops simply aren't prepared to spend that much more on their custom application development efforts.

Their conclusion was, broadly that HTML5 is now the most productive choice overall. Of course, this is about apps that need to run on mobiles as well as desktops. But is anyone writing an app in 2012 *not* thinking about mobile use?

### IDEs

Eclipse	Eclipse now supports HTML5: <a href="http://blog.webagesolutions.com/archives/419">http://blog.webagesolutions.com/archives/419</a>
Microsoft Visual Studio	JS and HTML5 are now Microsoft “first class citizens”: <a href="http://rww.to/xq3xHi">http://rww.to/xq3xHi</a>
NetBeans IDE	Full support for JavaScript, HTML and CSS3: <a href="http://netbeans.org/features/javascript">http://netbeans.org/features/javascript</a>

### Libraries and frameworks

H5BP	cross-browser abstraction: <a href="http://html5boilerplate.com">http://html5boilerplate.com</a>
Modernizr	cross-browser abstraction: <a href="http://www.modernizr.com">http://www.modernizr.com</a>
Ext JS	cross-browser widget library: <a href="http://www.sencha.com/products/extjs">http://www.sencha.com/products/extjs</a>
jQuery	extensive cross-browser JavaScript library: <a href="http://jquery.com">http://jquery.com</a>
Backbone	JavaScript MVC library: <a href="http://documentcloud.github.com/backbone">http://documentcloud.github.com/backbone</a>
Knockout	JavaScript MVVM library: <a href="http://knockoutjs.com">http://knockoutjs.com</a>

### Test tools and frameworks

JStestdriver	JavaScript test runner and console: <a href="http://code.google.com/p/js-test-driver/wiki/GettingStarted">http://code.google.com/p/js-test-driver/wiki/GettingStarted</a>
Jasmine	Behaviour-driven test framework: <a href="http://pivotal.github.com/jasmine">http://pivotal.github.com/jasmine</a>
Selenium	browser automation: <a href="http://seleniumhq.org">http://seleniumhq.org</a>

Table 2 – Some available HTML5 development tools



The final factor in developer productivity is the question of available skills. There is already a large pool of JavaScript developers out there and, as you would expect, the proportion of them that is HTML5-savvy is growing roughly exponentially (see Figures 5 and 6, below).

Developers who remain slow to learn JavaScript, HTML5 and CSS3 are simply going to lose out to the burgeoning horde of programmers with the right skills.

“The Web has not seen this level of transformation, this level of acceleration, in the past ten years... We’re betting big on HTML5”

Vic Gundotra, VP of Engineering, Google

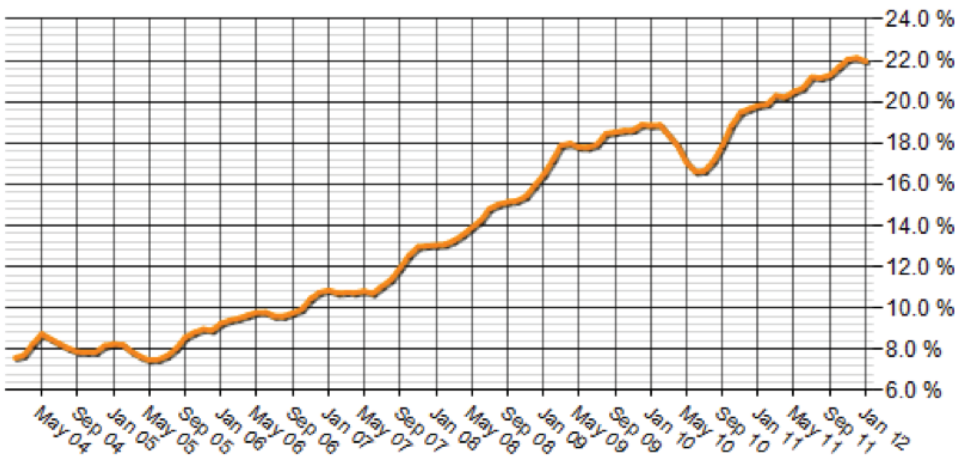


Fig. 5 – JavaScript skills as a proportion of all programming jobs advertised across the UK<sup>36</sup>

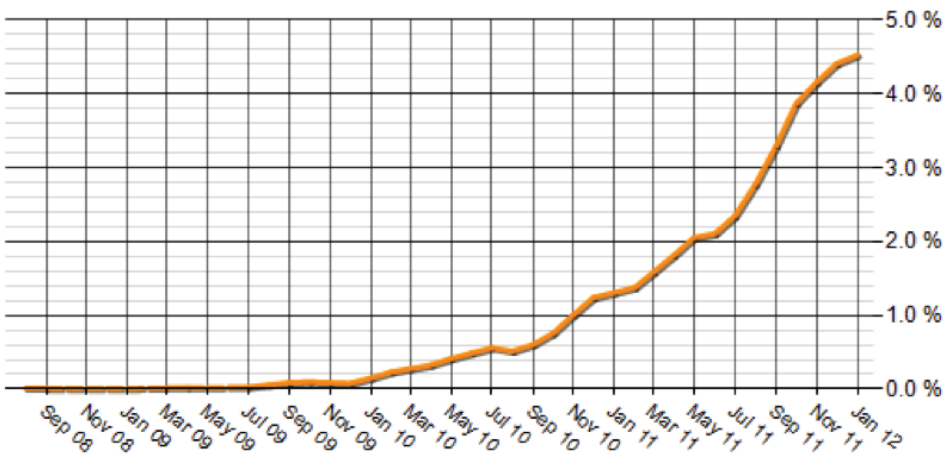


Fig. 6 – HTML5 skills as a proportion of all programming jobs advertised across the UK<sup>37</sup>

## ABOUT THE CAPLIN FRAMEWORK

The Caplin HTML5 framework for trading is built on top of the best HTML5 libraries and tools currently available. It integrates these and extends them where necessary to provide a complete development ecosystem.

On top of this we have built a layer of domain-specific business logic and a suite of trading components, libraries and interfaces to help you build awesome Web trading apps.

We constantly monitor the fast-moving world of HTML5, looking for newer and better pieces of the ecosystem. Where there are no tools available that can do a good enough job, we build our own – such as our class-loading and dependency management JavaScript bundler and our high-performing low-latency data-rendering framework. But we never stop looking for emerging standards, and if we identify one that's good enough, we substitute it for our own code.

For example, after a careful evaluation we recently replaced our event and data binding layer with Knockout, and we plan to retire our JS bundler once the ECMAScript Modules proposal is ratified.\*

We constantly contribute to open source tools projects to make them better for enterprise JavaScript and real-time applications, and we open-source our own tools where appropriate.\*\*

\* see <http://wiki.ecmascript.org/doku.php?id=harmony:modules>

\*\* for example, our dynamic-mock library:  
<http://mock4js.sourceforge.net>

## OVER THE EDGE

We are at the tipping point. The advantages of HTML5 are huge and obvious, and all the world's leading software companies are now committed to it. The providers of the plug-ins that served as stopgaps are frantically re-tooling for HTML5. All new browsers support it, it has been proven effective for even the most demanding business applications, and consumer adoption is soaring. It is supported on all tablets and smartphones. A major part of Microsoft's new Windows 8 operating system is based on it.

Many businesses have hesitated so far, as they wonder whether a sufficiently high proportion of their end-users have HTML5-compliant browsers; whether HTML5 is "ready" yet; whether the development tools available are sufficiently productive; and whether they can overcome the reactionary

pressure from developers who are reluctant to learn new skills. All these objections are about to go away, for the reasons outlined above.

This is the kind of situation where change, when it comes, comes like an avalanche. Or perhaps a better metaphor is a stampede, as mainstream application providers suddenly perceive en masse that the time has come, and race to catch up while, on the user side, the remaining stragglers are finally forced to upgrade their browsers because the old ones just don't work any more. It's a classic self-reinforcing cycle of rapid change.

That stampede will happen in the next few months, as it becomes universally apparent that none of the objections to HTML5 is valid any longer.

## AIMING HIGH

As the coming standard platform for Web delivery of applications, HTML5 is now almost universally endorsed in the software industry. I've argued that 2012 is the year in which it will pass the tipping point in the marketplace and become the dominant GUI development technology.

If you are choosing a Web application technology in 2012, going for anything other than HTML5 is a very risky decision, when even the providers of proprietary plug-in-based solutions are backing away from them.

If you decide on HTML5, you then have to make an important decision about toolkits and frameworks. For complex Web pages and simple Web apps, the existing general-purpose tools are already sufficient to provide an impressive level of productivity and maintainability. But more complex applications often need a higher-level framework that is tailored for their specific needs. Such domain-specific frameworks are now emerging in areas such as 3D visualization, gaming and animation.

At Caplin, we've spent years addressing this requirement in the case of high-end online trading, so this is where it gets personal. If you're building a Web-based trading app, we've already solved the framework problem for you (see box).

HTML5 is still new: there is a learning curve to climb, and pitfalls to avoid. At Caplin, we specialize in helping you climb the curve and avoid the traps when building financial apps, and we have a complete set of tools to do the job. If you're contemplating a project of this kind, call us. We can probably help.

## REFERENCES

1. HTML 5 Editor Ian Hickson discusses features, pain points, adoption rate, and more <http://tek.io/gilEHX>
2. Google Bets Big on HTML 5: News from Google I/O <http://oreil.ly/P5dlA>
3. Thoughts on Flash <http://www.apple.com/hotnews/thoughts-on-flash/>
4. Steve Ballmer says HTML 5 is the future <http://shar.es/fZLjP>
5. See for example: Microsoft Makes the Case for More jQuery, Fewer Dependencies <http://rww.to/qpRA7u> and Build 2011: HTML5 + JS 'First-Class Citizens,' Silverlight Takes a Back Seat <http://rww.to/prNc90>
6. The Internet Explorer 6 Countdown <http://www.ie6countdown.com/>
7. BBVA Gives Google Biggest Google Apps Contract To Date <http://bit.ly/zKcNJI>
8. How Secure Is Google Chrome? <http://soc.li/iL5eKUg>
9. German gov't endorses Chrome as most secure browser <http://shar.es/fwn45>
10. Google Wraps Internet Explorer in Chrome Clothing <http://bit.ly/rL2ciH>
11. Our plans to support modern browsers across Google Apps <http://goo.gl/j05xy>
12. <http://bit.ly/A7HNnY>
13. <http://bit.ly/wY4aUQ>
14. This is achieved via, amongst other things, a JavaScript scheduler that provides prioritized pseudo-threading without Web Workers, and a tunneling streaming protocol that is able to use a range of alternative mechanisms if Web Sockets are not available.
15. Building Mobile Apps? Start With Web; Move To Hybrid <http://bit.ly/zx9NJL> If you don't want to spend \$499 on the report, there's a useful summary of it here: Hybrid HTML5 Apps Are Less Costly to Develop Than Native <http://rww.to/xWMkry>
16. Google Bets Big on HTML 5: News from Google I/O <http://oreil.ly/P5dlA>
17. YouTube Begins to Support HTML5 <http://rww.to/zEhAq>
18. Thoughts on Flash <http://www.apple.com/hotnews/thoughts-on-flash/>
19. Steve Ballmer says HTML 5 is the future <http://shar.es/fw1CR>
20. Microsoft launches IE6countdown.com to track the death of Internet Explorer 6 <http://tnw.to/1CZ0z>
21. The Internet Explorer 6 Countdown <http://www.ie6countdown.com/>
22. Microsoft Windows 8 Preview With HTML5 And Javascript Apps <http://bit.ly/IsGrHT>
23. Financial Times drops iPad app, replaces with Web site <http://bit.ly/pH8WQg>
24. Financial Times Hits 1M Users On HTML5 Site That Dodges Apple's Tax <http://tcrn.ch/vbwqaX>
25. IEBlog Browsing Without Plug-ins <http://bit.ly/ndhSAH>
26. Our plans to support modern browsers across Google Apps <http://goo.gl/j05xy>
27. Build 2011: HTML5 + JS 'First-Class Citizens,' Silverlight Takes a Back Seat <http://rww.to/prNc90>
28. Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5 <http://adobe.ly/s8Anxl>
29. 10 years on Microsoft celebrate the death of IE6 <http://bit.ly/yqHh4L>
30. Browser Trends January 2012: Chrome 15 is NOT the World's Favorite Browser! <http://www.sitepoint.com/browser-trends-january-2012/>
31. Google Wraps Internet Explorer in Chrome Clothing <http://www.wired.com/wiredenterprise/2011/12/google-chrome-frame/all/1>
32. BBVA Gives Google Biggest Google Apps Contract To Date <http://bit.ly/zKcNJI>
33. Browser Trends January 2012: Chrome 15 is NOT the World's Favorite Browser! <http://www.sitepoint.com/browser-trends-january-2012/>
34. From analysis of <http://bit.ly/z0YnVs> and counting IE9, Chrome, Firefox, Safari 5.1 and Opera 11 as fully supporting HTML5 and CSS3 major elements (not including indexed databases, 3D CSS transforms and animations), IE8 as partly supporting them, and IE6 and IE7 as not supporting them.
35. When can I use... <http://caniuse.com>
36. ITJobsWatch <http://www.itjobswatch.co.uk>
37. ITJobsWatch <http://www.itjobswatch.co.uk>

Caplin's **Web Trading Technology** provides an HTML5-ready framework for building fully featured Web trading applications that run in a standard browser.

Our technology serves over 100,000 end-users at financial institutions that include ANZ, Barclays Capital, Crédit Agricole, Citi, MarketAxess, Nomura, RBS, Standard Bank, TD, UBS and UniCredit.

Our domain experience sets us apart from companies providing generic technology frameworks.

Caplin Systems is privately held and headquartered in the City of London.

## CAPLIN

### CAPLIN SYSTEMS LTD.

Cutlers Court, 115 Houndsditch, London EC3A 7BR, United Kingdom  
+44 20 7826 9600

### CAPLIN SYSTEMS, INC.

5 Penn Plaza, Suite 1982, New York, NY 10001, USA  
+1 (212) 835 1574

### CAPLIN SYSTEMS (ASIAPAC) LTD.

Room 1201, 12F, Shanghai Industrial Investment Building,  
48-62 Hennessy Road, Wanchai, Hong Kong  
+852 2987 7798