Sencha

The Development Manager's Quick Guide to HTML5

What Is HTML5 – and Why Should You Care?

Table of Contents

HTML 5 Is Making a Big Impact – and Quickly

HTML5 is the biggest leap forward in web technologies in the last generation. With broad support from Apple, Mozilla, Microsoft and Google, all the major browsers have rapidly incorporated HTML5 features. Apple Chairman Steve Jobs has said that the world is moving to HTML5 as the open-standard development solution for rich applications. And did you know that HTML5 is now one of the fastest trending searches on technical job sites?

But with people pointing to everything from simple JavaScript animations to CSS3 effects as examples of HTML5, it's not surprising that there is confusion about what HTML5 is and what it means. So…what exactly is HTML5, and why should development managers even care?

This paper provides a practical overview of HTML5 core technologies and explains why – in just a few short years – HTML5 has attracted the attention and support of industry leaders like Google, Apple and Microsoft. Based on Sencha's experience with HTML5 technologies, it summarizes the advantages of HTML5-based applications and discusses how frameworks and tools such as those offered by Sencha can facilitate the development of HTML5 applications.

HTML5 at a Glance

When the media refers to "HTML5," they typically mean the broad collection of next-generation web technologies that have been implemented in the current generation of web browsers. Some of these technologies are not strictly part of the HTML5 draft standard. For example, CSS3 have never been part of the HTML5 specification, and many others (such as Web Workers) were originally part of the spec but were later spun out as separate specifications.

At Sencha, we think the right way to refer to this collection of technologies is as "the HTML5 Family," but for convenience, we'll refer to them as "HTML5" in the rest of the paper. The family members of HTML5 (like all families) are in very different stages of maturity and implementation. Most are fully implemented in the latest revision browsers, but some are still in early draft and may be altered significantly before they are finalized.

The core World Wide Web Consortium (W3C) HTML5 spec is just one part of the HTML5 family of specifications. The full list of specifications includes:

- The Core HTML5 spec
- Cascading Style Sheets Version 3 (CSS3)
- Web Workers
- Web Storage
- Web Sockets
- Geolocation
- Microdata
- Device API and File API
- And more specifications in progress

(Note: We don't include SVG in this list because while it has been a standard since 2001, it still lacks broad and complete browser support.)

What's important to understand is that HTML5 originated as an initiative to transform the web browser into a richer application platform. The HTML5 mission was to enable web applications to be built with rich user interfaces, offline capabilities and deep but secure hardware access capabilities. Because HTML5 apps are web-deployed, they offer the benefits that the Web has always delivered - namely:

- A universal, cross platform client: the web browser
- Easy searchability and indexing (including deep linking)
- The ability to trivially include third-party services and mashups
- Zero-hassle deployments and updates (it's just on the Web)

Core Features of HTML5

HTML5 substantially changes many aspects of the HTML language, but most changes are invisible to end-users. These "user-invisible" changes include a new content model, a new layout model, and new accessibility features and browsing contexts.

Other core features of HTML5 include the following:

- In many cases, developers can use HTML5 to do what they currently do with JavaScript, or server workarounds. For example, HTML5 adds semantic tags for many common content elements (for example, a form field for email address

A Brief History

HTML stands for HyperText Markup Language –a "simple markup language used to create hypertext documents that are platform independent." But the "simple markup language" was so useful that over time it quickly grew in complexity, and was soon accompanied by a style sheet language (CSS) and a standard scripting language (JavaScript).

By 2005, the web was already an application platform, but one that was primarily based on server-side code execution with extremely simple JavaScript code executing the client-side. The rise of AJAX opened the door for the browser to become a full-fledged client-side platform. In short order, the proliferation of intricate, client-side web applications such as Google Gmail and Yahoo! MapQuest showed that the web could deliver interesting, complex, globally available applications. But in this pre-HTML5 world, plug-in technologies such as Flash and Google Gears had to be used to make up HTML4's feature deficiencies so that developers could deliver rich experiences.

In 2008, the W3C and WHATWG web standard consortia jointly published the first public working draft of HTML5. Although the specification work is ongoing, browsers have already implemented many HTML5 features. On mobile, this has been accelerated by the widespread use of the WebKit layout engine by the mobile browsers on iOS, Android, Blackberry, WebOS and Bada.

entry). It also includes new markup for menus and navigation sections that result in cleaner, human-readable HTML; as a result, developers can replace the current blizzard of <div> tags with meaningful markup, such as <nav> and <aside>.

- For forms, HTML5 adds support for PUT and DELETE form actions, which will simplify server-side processing. It also provides native support for adding form elements dynamically (something that currently has to be done in JavaScript).
- For users, the highest-impact change in HTML5 is the addition of audio (<audio>) and video (<video>) tags and an immediate mode 2D bitmap drawing format (<canvas>). HTML5 audio and video tags allow playback without the use of plugins, while canvas supports rich 2D drawings.
- HTML5 removes many of the presentational markup elements that litter earlier HTML specs, like <center> and <font>. It also disallows direct table styling; instead, developers must use CSS.
- HTML5 officially eliminates frames except for the <iframe> inline frame.

Other features in the core HTML5 spec include drag-and-drop of web content across the application's canvas, drag-and-drop of files from desktop to application, support for cross-document messaging, persistent content caching directives, and support for user-editable content.

The core HTML5 capabilities are, by themselves, a strong foundation upon which to build rich web content.

HTML5 and Its Relationship to CSS3

What many people think is "HTML5" is actually CSS3. CSS3 is a collection of sub-specifications that work in concert with the document structures created with HTML5. In most cases, HTML markup defines the structure of web content; JavaScript manages behavior; and, CSS determines the content's visual presentation.

CSS3 modules have an impact on just about every aspect of presentation, from element transformations (such as rotation and scaling) to advanced image-based borders. CSS is even used to deliver font files that can be included dynamically in webpage content. Developers can even use CSS3 to specify a style per device – for example, to differentiate styling between TVs, mobile devices, and print views using CSS3's @media rules. They can also apply the same aesthetics to particular page elements by finding and applying styles using CSS3's "selectors" module.

Some of the most exciting CSS3 modules for application developers are "Transitions" and "Animations." These style declarations govern how page element styles change over time when those elements' properties are changed.

For example, the Transition declaration allows developers to specify what happens when a user hovers their mouse over a link on their site – for example, the link's textual color and font size could change to a new style over a second. CSS Transitions can be added dynamically on the fly by rewriting or applying new CSS classes using JavaScript.

Animation rules, on the other hand, allow for multiple changes to CSS properties values over time. Animations can iterate finitely or infinitely; either way, the browser automatically calculates visual effects transitioning between each state. On mobile, CSS Animations are often accelerated on the device's GPU, resulting in an enormous performance gain when compared to a plain JavaScript animation.

Another exciting feature of CSS3 is support for richer layouts. Flexible and fluid, CSS3 layouts, such as the Multi-Column Layout and Flexible Box Layout, define new CSS properties for the creation of complex content flows without having to resort to highly nested <div>'s or tables with head-scratching math calculations.

CSS3 has many other modules, and it promises to be a vital tool in web site and application design; put simply, it's changing the way developers visually communicate web content. For users, the various modules work together to create a richer web experience. For developers, content is easier to style, without resorting to image effects. CSS3 works in expected ways across the wide variety of devices used to access the Web today.

Why Use HTML5 for Application Development?

We're very excited about the HTML5-based development here at Sencha because it promises richer applications and more developer power tools, and we've been eager to take advantage of new HTML5 features as they have arrive in the browser. But for companies that are already happily using server-based web applications, it's worth asking, "Why use HTML5 at all?"

The short answer is that you can create richer, more responsive application experiences that your users can take offline with them. We lay out the long answer in the following sections, where we'll explore the key advantages of HTML5 and discuss why it may provide the right tool for developing your next great application.

A Web Experience that Uses Open Standards

For years, plug-ins like Flash and Silverlight enabled developers to compensate for the Web's lack of rich features. However, this is no longer necessary with HTML5, which is excellent news given the many limitations associated with plug-in technology. Apart from well-publicized mobile platform performance problems, plug-ins have introduced stability and even security problems.

HTML5 web experiences are built on an open standard. The developer's audience is no longer limited to those who chose to allow the download of a plug-in or devices that have plug-in support – a major issue in the case of some mobile devices. Instead, any browser client, including any mobile device browser, can deliver the same rich Web experience.

Rich Web Experiences with Cross Platform, Cross Device, Cross-Browser Consistency

HTML5 specifies how browser clients should interpret HTML markup, making consistency an implicit feature of HTML5. As previously mentioned, browsers supporting most HTML5 features are currently in use on all major desktop and mobile device platforms. As long as a browser is HTML5 compliant, a developer can create a single app and distribute it consistently across all combinations of browsers, devices and platforms.

A Rich Web Application, Not Just a Web Site

As previously mentioned, HTML5 was created to enable the development of richer web applications, but it's worth exploring why developers turned to the Web as a platform in the first place.

The initial attraction of the Web was ubiquitous access; instead of having a native application tied to whatever machine it was installed on, users could access content from any browser anywhere in the world at any time. With web applications, users gained the freedom to interact with their favorite applications anywhere.

Web application development was also easier compared to native application development. Native application development uses separate languages for each platform; this cumbersome process results in an extensive (and often expensive) ramp-up. In contrast, Web application development done with HTLM5 offers a far easier development process.

Today, there are a number of great developer tools that make the process even easier for developers seeking to create rich web applications; for example, Sencha offers a user interface builder called Sencha Designer that gives developers a visual drag-drop design system, enabling them to quickly and easily build rich web applications.

A Rich Web Application for Mobile Devices

With the growing ubiquity of mobile devices, people are now accustomed to accessing the Web on-the-go using phones, tablets, and other devices. What used to be a desktop-only activity is now a 24/7 anytime, anywhere possibility.

Due to the lack of Flash support on market-critical mobile platforms (such as iOS) and alternative technologies for creating rich, mobile-optimized Web experiences, many developers that want to create rich mobile application experiences end up creating native mobile applications. However, native apps distribution has a significant and potentially pricey caveat: to distribute a native app, developers must submit it to an app store ecosystem. The native app store ecosystems, particularly the Apple App Store, are notorious for having lengthy approval processes and arbitrary rejection of applications. On applications that make the cut, they also charge a percentage of sales to app developers as a fee.

Before HTML5 matured and gained widespread adoption by major browser clients, native apps were the only option for mobile applications. But today, with the help of HTML5, developers can use the Web as both a desktop and a mobile platform. With HTML5-based mobile web applications, developers can control application distribution and avoid app store constraints. In addition, developers can create rich web applications that have shorter, more flexible development cycles, delivering a consistent user experience to any mobile browser client, regardless of the platform.

Let's elaborate a little further on this last point: mobile Web applications are far more flexible to develop and update than their native counterparts. For example, developers can push bug fixes and feature upgrades immediately and seamlessly – with no required action by the user. This is in direct contrast with native application development cycles; in native apps, application changes must be consolidated into downloadable version updates, which platform owners seek to minimize in order to reduce the number of updates that users must be bothered with. This, in turn, delays developers getting new functionality to users. Developers also have to upload upgrades to a mobile app store and queue for approval from the app store. At this point, the developer can only wait; the application is reviewed at the app store's availability and discretion, often resulting in weeks and even months of waiting. Updates also run the risk of being rejected, and if it is, the developer must edit the application based on the app store's comments and go through the entire process again. If and when a developer is able to promote an update to the app store, there's no guarantee it will reach its intended audience; users must actively approve and download the update.

But perhaps the biggest reason to develop mobile apps using HTML5 is its ubiquity and consistency. With HTML5, developers can create rich mobile Web content that can be distributed simultaneously and consistently to all mobile platforms. This means, for example, that Android, iOS and Blackberry OS web applications can be developed and released simultaneously.

Advances in the mobile web process would be meaningless if it were not possible to easily develop interactive experiences on the mobile Web. Gesture detection and fixed-position toolbars are highly desirable features, but they are not part of the HTML5 specification. This is where toolkits and frameworks can help. Similar to desktop-side tools and frameworks, HTML5 technology-based mobile Web toolkits and frameworks are now available to support developers. For example, the Sencha Touch mobile Web framework makes it possible for developers to detect gestures and have fixed position toolbars and deliver video and audio for the mobile web – all without plug-ins. (To get a better idea of what is possible today with mobile web frameworks, check out these Sencha mobile Web examples on your favorite smartphone or tablet at www.sencha.com/products/touch/demos/.)

Rich Applications with a Shorter Development Cycle and Fewer Resources

HTML5 also offers developers instant gratification. HTML5, CSS and JavaScript are interpreted languages that require no compilation cycle. This means developers can debug in-browser, make a quick edit, and refresh their browser window to see changes live. This contrasts starkly with traditional native or plugin development with Flash, Silverlight or Objective C; edits can't be viewed instantly, but rather only after a lengthy compilation and deployment process. And if the edit is incorrect, the developer has to go through the entire process all over again.

So…Is HTML5 Right for You?

HTML5 is equipped with many great features that will modernize the Web by incorporating technologies once supported only through third-party plug-ins. But is it right for you?

Are you planning to develop an application with limited resources?	**HTML5 is a good fit;** HTML is human readable, and there are over ten million web developers (compared to much fewer native and plug-in developers).
What audience are you targeting? Are you planning to develop an application for a multi-platform audience with access anytime and anywhere?	**HTML5 is a good fit;** with HTML web-based applications, your application is reachable from anywhere at anytime.
Are you planning to reach your audience on the go with devices such as smartphones and tablets?	**HTML5 is a good fit;** with HTML web-based mobile applications, your application consistently displays across mobile web browsers, including future devices.
Are you looking to reach mobile audiences, but without having to go through mobile app store ecosystems?	**HTML5 is a good fit;** with HTML web-based mobile applications, you don't even need the approval of a app store to reach your audience.

With HTML5 and some great support tooling, it is now possible to create a sleek, modern web application that is accessible from any browser, any desktop, and any mobile device, anywhere at any time. Moreover, applications can be created quickly and easily without a lot of experience.

How Sencha Helps Ensure Success with HTML5

Our mission at Sencha is to create the frameworks and tools that empower developers to create amazing cross-browser application experiences that execute with precision and performance.

Sencha creates application frameworks to enable developers to build cross-platform, cross-browser Web apps. Our product portfolio consists of two classes of products: application frameworks and tools. The frameworks are the core systems that enable the rapid creation of true, cross-browser web applications, from IE6 to Chrome 10 and everything in between. Our tools accelerate the developer's workflow even further by leveraging the frameworks to enable visual development of user interfaces and interactions.

Our main framework offerings are Ext JS and Sencha Touch:

- **Ext JS** is designed for the creation of desktop Web sites, enabling developers to build rich Web applications with ease. Ext JS also includes a large library of UI components and default UI themes that enable developers to focus on building the application they want.
- **Sencha Touch** is a framework used to build mobile and touch Web applications. It's the world's first application framework built specifically to leverage HTML5, CSS3 and JavaScript for the greatest power, flexibility and optimization.

Sencha products make specific use of HTML5 to deliver components such as audio and video, as well as a local Storage proxy for saving data offline. We have also made extensive use of CSS3 in our style-sheets to provide the most robust styling layer possible.

To complement the frameworks Sencha offers, we also make available two tools: Ext Designer and Sencha Animator (beta):

- **Ext Designer** is a desktop application that helps you create interfaces faster than ever in an easy-to-use, drag-and-drop environment. Using Designer, companies can create new designs quickly, enabling Ext JS developers and product designers to develop the user interface for their web app in a reduced amount of time.

- **Sencha Animator** is a desktop application used to create animations for mobile browsers and touchscreen mobile devices. Motion designers can use Animator to create rich experiences for today's most popular devices using just CSS3 and no plug-ins.

Learn More

If you have any questions about HTML5 or want to know more about how Sencha can help you leverage HTML5 to improve Web experiences, visit us online at www.sencha.com. You can also:

- Mingle with other web developers and Sencha experts on our developers forums (www.sencha.com/forum)
- View screencasts and code demos at our Learning Center (www.sencha.com/learn)
- Sign up for training sessions on our frameworks and tools to jump-start new developers as they use HTML5 (www.sencha.com/training)

1700 Seaport Boulevard
Suite 120
Redwood City, CA 94063
1 888 736 2421
www.sencha.com
@sencha

Sencha makes application frameworks that equip developers to create, deploy, and optimize compelling experiences using web-standard technologies. The company's flagship products, Ext JS and Sencha Touch, are a cross-browser JavaScript frameworks which produce rich internet applications for desktop and mobile devices. More than one million developers worldwide—representing more than 150,000 companies—use the Sencha family of products to build amazing applications every day.