

Look ahead to emerging web technologies in 2011

20 years after its inception, the web is still developing

Skill Level: Introductory

[Martin Streicher](mailto:martin.streicher@gmail.com) (martin.streicher@gmail.com)

Software Developer
Pixel, Byte, and Comma

11 Jan 2011

At 20 years old, the World Wide Web remains abuzz with innovation. Now more than ever, implementation trumps invention. Developers can expect to spend more time creating and less time coding in the new year.

December 2010 marked the twentieth anniversary of the creation of the World Wide Web. In its birthday suit, the web was composed of a single computer—a NeXT workstation, to be specific—running both a rudimentary web server and a nascent browser. Two decades later, that one device has become one hundred million devices or more, ranging from traditional desktop computers to wireless handsets to automobile dashboards. If humans had evolved at the same pace as the web, cave drawings might have depicted how to lightly braise woolly mammoth in shallots and red wine.

Frequently used acronyms

- **API:** Application programming interface
- **HTML:** Hypertext Markup Language
- **HTTP:** Hypertext Transfer Protocol
- **REST:** Representational State Transfer
- **SMS:** Short Message Service
- **SQL:** Structured Query Language
- **UI:** User interface
- **URI:** Uniform Resource Identifier

Twenty years later, the pace of innovation on the web isn't slowing. HTML5, although not yet ratified, is being adopted almost universally and rather quickly, driven by emergent support in most major browsers. Competition between JavaScript engines and libraries is yielding better interactivity for users, better tools for software developers, and even new forms of highly scalable computing. And, of course, some new, clever website makes its debut each week, proving that the web remains a remarkable platform to realize ideas.

However, the lion's share of inventiveness on the web today isn't tied to new technology. Instead, the current emphasis is on implementation. Tools and techniques change rapidly, and developers experiment and eviscerate bad ideas, refine good ones, and canonize the best notions as best practices.

Web development in 2011 is to be *evolutionary*, not *revolutionary*—about *tools*, not *technology*. Several trends are sure to appear or be bolstered in the coming year.

Look, mom—no root prompt!

Cloud computing has been widely hyped as the cure for the common cold and even better than whole-grain sliced bread. Perhaps those claims exaggerate its potential, but not by much. Cloud computing has had a remarkable impact. Services like Amazon Elastic Compute Cloud (Amazon EC2), the Rackspace Cloud, and other similar virtual server farms have made the machine room and the need for capital investments obsolete. Turning a machine on is as simple as clicking the mouse.

IBM and the cloud

Dig deeper into cloud computing and check out IBM cloud offerings at [developerWorks Cloud computing](#).

Further, turnkey hosting providers such as Heroku provide clouds with silver linings: application hosting with no system administration. Just install your code and go. Heroku is especially notable because it has fomented a cottage industry of specialized cloud providers that complement its core services. Need search? Plug in Websolr. Global SMS? Use Moonshado. Prefer a "no SQL" database? Choose from Mongo, Redis, or JasonDB.

So long, root prompt! It's been nice knowing you. With little or no systems administration, developers will be free to focus on application code in 2011.

Application, phone home

As Tim Berners-Lee wrote, even the newborn web "demonstrated a profound

concept: that any person could share information with anyone else, anywhere." Indeed, given the web, a browser, and a little handle called a URI (the now-familiar "dub dub dub" notation), one can instantly find the current weather in Mumbai, read about the challenges and victories of fellow cancer patients, or access the complete works of Shakespeare, among countless other possibilities. Berners-Lee and others' HTML and HTTP have served their purposes well. See [Resources](#) for a link to his article.

But the web is no longer just hypertext documents. Fundamentally, it is no longer "read only"; in fact, the browser is more engine or operating system than document viewer. Pages, feeds, and downloads have been replaced with tweets, callbacks, and synchronization. The expectation is that new content is delivered as soon as it becomes available.

To date, only a relatively small number of applications have taken advantage of this dynamism. Recall Apple's Push service and the failed but nevertheless groundbreaking Google Wave. Otherwise, the technology has been too complex or proprietary to make inroads.

Yet the barriers are falling. HTML5's sockets add two-way communication to the browser as well as frameworks that are cropping up to standardize server implementations. PubSubHubbub, too, precludes the need to poll for new content. Simply register your interest in a feed using PubSubHubbub, and the server calls back whenever the feed is updated. The open source Diaspora project, a decentralized Twitter, also promotes the notion of subscriptions.

In the wired, wired, wired world that is 2011, sites will stay in touch with you.

Just when you thought it was safe...

For years, developers have wrestled with support for multiple platforms. Recall coding shrink-wrapped software for Mac and Windows® (in all its flavors); porting scientific programs to different incarnations of UNIX® (and 18 flavors of windowing systems); and building web pages for Windows Internet Explorer® version 6 and almost every other browser. These hassles pale in comparison to the recent explosion of incompatible devices. Count Mac OS X, Apple iOS, Google Android, RIM Blackberry devices, the standard desktop, and Internet Explorer among them. At times, it's difficult to discern whether so much competition is a blessing or a curse.

The good news is that robust cross-platform development tools do exist. For example, Rhodes is a Ruby on Rails-like framework that generates applications for a number of devices. Similarly, Appcelerator Titanium combines JavaScript and HTML to build native applications for portable devices. Throw in Ruby, PHP, or Python, and you can also use Titanium for desktop applications. Other nascent options—PhoneGap and MoSync—are available, and most are offered under liberal

open source licenses.

In general, each solution produces dynamic, embedded websites for the device. Many of the frameworks integrate with features like geolocation, making the application seem truly native. jQuery Mobile is a recent addition to the cross-platform corpus of tools and also generates an embedded website.

Software application developers tend to specialize in a single platform. If your platform is the Web, such cross-platform tools can help you conquer the rest of the Wide World.

R.I.P. JavaScript

Admittedly, it's premature to eulogize JavaScript. The language continues to enjoy a renaissance, even as some long for its untimely demise. Most developers either love JavaScript or hate it, perhaps depending on their tolerance for pain, the amount of caffeine consumed, or the task at hand. Ideally, JavaScript will be replaced with another language that is less quirky. This is unlikely to occur in the coming year (or any time soon), and Nostradamus is strangely quiet on the whole topic.

If you find yourself in the "hate it" camp, take solace in polished JavaScript toolkits, such as jQuery and Prototype, and clever dialects, like CoffeeScript. The CoffeeScript site describes the project as "an attempt to expose the good parts of JavaScript in a simple way," since "underneath all of those embarrassing braces and semicolons, JavaScript has always had a gorgeous object model at heart."

CoffeeScript compiles into printed, readable JavaScript; it can be included directly in the browser, where it compiles and runs in situ. CoffeeScript syntax is easy to learn and gaining in popularity. It's like adding cream and sugar to a bitter cup of joe.

Location, location, location

The real estate proverb that it's all about "location, location, location" is an apt mantra for web development in the immediate future. With browsers unchained from the confines of the desktop, developers have the opportunity to build applications that instantly affect people's lives. Want to find the closest metro stop, or the best Thai restaurant in the neighborhood? Or perhaps how much every house on your street sold for?

Geolocation, derived from either Global Positioning System (GPS), cellular signal triangulation, or IP address, is near-ubiquitous, so there's little reason to ignore it. Indeed, ignore geolocation at your own peril.

Synchronization, synchronization, synchronization

In addition to the variety of devices used to access the web, a sheer multitude of devices are now online. It was quite common in 2010 for an individual to carry a laptop computer, a cell phone, and perhaps an e-Reader, each with connectivity and unique features and purposes. But to paraphrase Spider-Man, with great convenience comes great complication. That is, if you use the same application on several devices, how can you keep the information in sync?

If the device remains connected, a site or service can house the data. Otherwise, the data must be synced opportunistically and, ideally, automatically. Apple MobileMe provides seamless syncing (which is one of the service's best selling points), but the service is proprietary to Mac OS X and iOS and requires an annual subscription. Some developers create their own tools to synchronize content and data, including offline "widgets" that serve as proxies for connectivity. Simply enter your data locally and sync whenever a connection is available. The Harvest time management application is one example. The Apple iPhone version of the application retains hours worked until a connection is made, when it transfers everything to its web service.

Consistency is increasingly crucial. Again, HTML5 offers new features for local storage, but stand-alone applications will likely prefer an "incrementally consistent" database, where local and remote instances can replicate as peers back and forth. CouchDB is one such database, as is Riak. A RESTful API is an additional benefit of the two.

If your user is constantly changing location, location, location, consider the tools for synchronization.

Put the "user" back in "user interface"

Finally, given all of these tools, a web application need not resemble the DOS prompt. Fine typography on the web became a reality last year, so there is no shortage of best practices and source material for beautiful and functional navigation, information architecture, forms, and interactivity. The browser, although now stretched far beyond its original intent, remains resilient.

But the browser is just one medium, and an application that works well with a browser and attached keyboard and mouse might not translate universally. For example, hover is useless without a mouse, and drag-and-drop is difficult for a small screen or users with large thumbs.

Product managers, developers, and designers would be wise to invest extra effort in making applications simple, direct, concise, and conformant to the capabilities of

each target device. Design for the intended audience—and device.

The web? What's that?

In a sense, it's something of a misnomer to continue to label web applications as such. More and more, applications and information are accessed via other platforms and devices, either directly and indirectly. It might be more appropriate to call web development *Internet development*, *service development*, or just *application development*. How the utility is consumed is increasingly irrelevant.

Yet like in years past, application development continues to require mastery of many disciplines. Data structures are still important; parsing strings is a common task; careful and thoughtful interaction design and presentation are mandatory. As application development circles again to many devices and many operating systems, choosing the right tools for the task is an invaluable skill. In 2011, you will have a great suite to choose from. No root prompt required.

Resources

Learn

- According to its home page, the [Amazon EC2](#) "presents a true virtual computing environment." Booting a machine requires a click of the mouse.
- The [Web development zone](#) specializes in articles covering various web-based solutions.
- Browse the [technology bookstore](#) for books on this and other technical topics.
- Stay current with developerWorks' [technical events and webcasts](#).
- Read Tim Berners-Lee's article "[Long Live the Web: A Call for Continued Open Standards and Neutrality](#)."
- Watch [developerWorks on-demand demos](#) ranging from product installation and setup for beginners to advanced functionality for experienced developers.

Get products and technologies

- [Appcelerator Titanium](#) is a free and open source application development platform for creating native mobile, tablet, and desktop applications using existing Web technologies.
- If you like JavaScript but dislike its syntax, try [CoffeeScript](#), which exposes the good parts of JavaScript in a simple way.
- [Diaspora](#) is an open source, decentralized social network that emphasizes privacy. Anyone can host a Diaspora server, or *seed*, and information is shared only with seeds you authorize. The code for Diaspora is hosted on [Github](#).
- [Google Wave](#) is a real-time collaboration platform for sharing messages and media.
- A dedicated Ruby hosting platform, [Heroku](#) offers virtual computing resources with no systems administration.
- [jQuery Mobile](#) builds a unified UI across all popular mobile device platforms. It is built on the [jQuery](#) and [jQuery UI](#) foundation.
- [MongoDB](#) (the name is derived from "humongous") is a scalable, high-performance, open source, document-oriented database.
- The oddly named but effective [PubSubHubbub](#) is a simple, open, server-to-server publish/subscribe protocol extension to Atom and RSS. Once you register your interest in a feed (subscribe), PubSubHubbub contacts you when the feed has new content available (publish).
- [Moonshado](#) is a simple, open, server-to-server web-hook-based publish and subscribe protocol that extends Atom and RSS.

- The [Rackspace Cloud](#), much like Amazon EC2, provides instant-on, virtualized computing resources, including hosts, file storage, and turnkey websites.
- [Redis](#) is an advanced key-value store. It is similar to [Memcached](#), but data is not volatile and values can be strings, lists, sets, and ordered sets.
- Programmers can rapidly write native smart phone applications with [Rhodes](#).
- The [Websolr](#) service hosts Solr indices. APIs are available for Windows .NET, PHP, Python, and Ruby.
- [Download IBM product evaluation versions](#) or [explore the online trials in the IBM SOA Sandbox](#) and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

Discuss

- [developerWorks blogs](#): Check out our blogs and get involved in the [developerWorks community](#).
- Create your [developerWorks community profile](#) today and [setup a watch list](#) on web development. Get connected and stay connected with [developerWorks community](#).
- Find other [developerWorks members interested in web development](#).
- Share what you know: [Join one of our developerWorks groups focused on web topics](#).
- Roland Barcia talks about [Web 2.0 and middleware](#) in his blog.
- Follow developerWorks' members' [shared bookmarks on web topics](#).
- Get answers quickly by visiting the [Web 2.0 Apps forum](#).
- Get answers quickly by visiting the [Ajax forum](#).

About the author

Martin Streicher



Martin Streicher is a freelance Ruby on Rails developer and the former Editor-in-Chief of [Linux Magazine](#). Martin holds a Master of Science degree in computer science from Purdue University and has programmed UNIX-like systems since 1986. He collects art and toys. You can reach Martin at martin.streicher@gmail.com.