



ELEMENTS

NOVEMBER 20, 2014

THE GROUP THAT RULES THE WEB

BY PAUL FORD

COURTESY THE W3C

You might have read that, on October 28th, W3C officially recommended HTML5 (<http://www.w3.org/TR/html5/>).

And you might know that this has something to do with apps and the Web. The question is: Does this concern you?



The answer, at least for citizens of the Internet, is yes: it *is* worth understanding both what HTML5 is and who controls the W3C. And it is worth knowing a little bit about the mysterious, conflict-driven cultural process whereby HTML5 became a “recommendation.” Billions of humans will use the Web over the next decade, yet not many of those people are in a position to define what is “the Web” and what isn’t. The W3C *is* in that position. So who is in this cabal? What is it up to? Who writes the checks?

The Web is a Millennial. It was first proposed twenty-five years ago, in 1989. Six years later, Netscape’s I.P.O. kicked off the Silicon Valley circus. When the Web was brand new, many computer-savvy people despised it—compared to other hypertext-publishing systems, it was a primitive technology. For example, you could link from your Web page to any other page, but you couldn’t know when someone linked to your Web page. Nor did the Web allow you to edit pages in your browser. To elite hypertext thinkers and programmers, these were serious flaws.

The Web was, however, very easy to set up and learn. It contained the seeds of its own transmission—anyone could learn HyperText Markup Language by reading a Web page then viewing the raw HTML beneath. The Web was made up of simple documents and images that linked to other simple documents and images.

The religion of technology is featurism, however, and, so, people began adding everything they could to the Web. How about displaying things in 3-D? How about text that blinks or text that scrolls across the page as a marquee? What about turning every single Web page into software? Different browsers—with names like Mosaic, Netscape, Internet Explorer, Cyberdog, Spyglass, Lynx, and Amaya—appeared, each carving out its own cultural and market niches.

With that complexity came Balkanization. Imagine that your Web browser only renders photographs in one format, and mine in another, and I send you a link to an image—you wouldn't be able to see it. Instead of one Web, you would have many. Anarchy would ensue and photographers would complain and complain.

As this Balkanization was beginning to happen, people realized that there was a need for a group to decide on a common language that would include all the necessary features. Then that group would need to write a document that contained every aspect of the evolution of hypertext markup language. This is the standardization process—technical diplomacy in the interest of commerce—and it is essential to the progress of the Internet. It is also not original to computing.

Consider the Buffalo Convention, of 1908, when player-piano manufacturers met at the Iroquois Hotel in Buffalo. At issue was the number of perforations per inch that would be punched into the rolls used to map out songs for the pianos; some people favored nine, some favored eight, and the difference meant increased costs, manufacturer distress, and customer confusion. In “Gathering of the Player Men at Buffalo

(<http://www.mmdigest.com/Gallery/Tech/MTR-1908-12-08-v47-n24-p31.pdf>),” the *Music Trade Review* described a heady scene in which Mr. P. B. Klugh, speaking for the Cable Company, said that it had adopted “the nine-to-the-inch scale” and that “they were not open to argument on the subject, as such a scale had given entire satisfaction.” Swayed, the manufacturers resolved the issue in favor of Klugh. As a result, we now live in a world where nine-holes-per-inch piano rolls are the standard. You would be a fool to build a player piano to any other metric.

Of course, the Web page is far more complex. It requires dozens of standards, governing words, sounds, pictures, interactions, protocols, code, and more. The role of Web parliament is played by the W3C, the World Wide Web Consortium. This is a standards body; it organizes meetings that allow competing groups to define standards, shepherding them from a “working draft” to “candidate recommendation” and “proposed recommendation,” and finally, if a standard has been sufficiently poked and prodded, granting the ultimate imprimatur, “W3C recommendation.”

The W3C has been meeting for twenty years (<http://www.w3.org/20/>), led by its director, Tim Berners-Lee, the principal creator of the Web. Its membership is drawn from close to four hundred academic, not-for-profit, and corporate organizations. Among its most engaged participants are large companies that build Web software and host enormous Web sites—ones like Google, Microsoft, and Facebook. They all pay dues (<http://www.w3.org/Consortium/fees?countryCode=US&quarter=10-01&year=2014#results>) for spots at the table—sixty-eight thousand five hundred dollars a year for the biggest U.S. firms, although not-for-profits and smaller firms pay far less, and less-prosperous nations adhere to a sliding scale.

The cultural mission of the W3C is to make the Web “available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or

physical or mental ability.” The way it accomplishes this is by committee, via standards documents.

If you want news about the development of the Web, you can visit the W3C home page (<http://www.w3.org/>) and scan the most recent news. Reading through the standards, which are dry as can be, you might imagine that standardization is a polite, almost academic process, where wonks calmly debate topics like semicolon placement. This is not the case. Important standards are sometimes forged in polite discourse, and sometimes in a crucible of tribal rage, leaving behind a trail of open letters, back-channel sniping, and high-dudgeon blog posts.

This is not some secret shame; it is an expected part of a healthy process. “Technology standardization is commercial diplomacy,” wrote Stephen R. Walli, a business-strategy director at Hewlett-Packard and a veteran of many such efforts, in a paper on the subject (<http://stephesblog.blogs.com/papers/stdsprimer.pdf>), “and the purpose of individual players (as with all diplomats) is to expand one’s area of economic influence while defending sovereign territory.”* Or, as Charles F. Goldfarb—who co-created a forerunner to HTML called Standard Generalized Markup Language, in 1974—once delicately put it (<http://lists.w3.org/Archives/Public/w3c-sgml-wg/1996Oct/0232.html>), on an e-mail list: “Multi-year projects in a highly political arena with changing personnel contributes to a loss of focus.” Which is to say: standards, like laws, emerge from fundamental conflict.

Since its first iteration, HTML has defined a set of rules for adding markup to textual content. If you wanted something to be a headline, you’d add `<h1>` tags around it: `<h1>Your Headline</h1>`. The `<h1>` is the markup. The “Your Headline” is just character data. Your browser, programmed to interpret the rules of HTML, would show it in an appropriately large format.

That's HTML at its essence: just a bunch of tags.

But, with HTML5, the markup language has become a connective tissue that holds together a host of other technologies. Audio, video, pictures, words, headlines, citations, open-ended canvases, 3-D graphics, e-mail addresses—it lets you say that these things exist and gives the means to pull them into one solitary page. You can even “validate” a page. At this writing, for example, Apple.com has one HTML5 error (<http://validator.w3.org/check?uri=apple.com&charset=%28detect+automatically%29&doctype=Inl>

That's pretty good: the New York *Times* has a hundred and forty-one.

Validity, in this scenario, is an ideological construct. The promise is that by hewing to the rules put forth by the W3C, your site will be accessible to more people than would a less valid page. Both pages work fine for most people; browsers are tolerant of all sorts of folderol. The ultimate function of any standards body is epistemological; given an enormous range of opinions, it must identify some of them as beliefs. The automatic validator is an encoded belief system. Not every Web site offers valid HTML, just as not every Catholic eschews pre-marital sex. The percentage of pure and valid HTML on the web is probably the same as the percentage of Catholics who marry as virgins.

The conflicts that led to HTML5 were more pronounced and public than usual. To understand why, you must go back to June, 1996, when a new version of HTML appeared. HTML3.2 was a big release, because it made official what had previously been practice. A language with the misbegotten, marketing-driven name JavaScript had been added to Web browsers; now every element on a page could come alive. HTML3.2 didn't actually say much about JavaScript, just something like, “In the future, there will be scripts in HTML.” And boy, were there.

“The by-design purpose of JavaScript,” one well-informed commenter

(<http://programmers.stackexchange.com/questions/221615/why->

do-dynamic-languages-make-it-more-difficult-to-maintain-large-codebases) wrote, “was to make the monkey dance when you moused over it.” These dancing monkeys eventually begat more dancing monkeys, first evidenced in things like pop-up windows, then later—with a major assist from Microsoft, which added a technology to Internet Explorer that made it possible to load in new data without refreshing the browser—in the form of “web apps” like Google Maps, Gmail, Twitter, and Facebook.* Now the whole Web is dancing monkeys. We still call Web pages “pages,” but many of them are actually software applications—“apps”—as complex to engineer as any word processor or video game. (Often, they *are* word processors, such as Google Docs, or video games, such as HexGL (<http://hexgl.bkcore.com/play/>).)

By 2004, this change, from page to app, was of great interest to global corporations, which saw this new, active Web as a potential source of tremendous profit. The W3C, though, was many years into an attempt to reengineer the Web. It was pushing for an *accessible* Web—one that worked for blind and disabled people, in which pages described their own contents. It also wanted a semantic Web, doing for ideas what it had done previously for documents—linking all of the world’s information into a sort of decentralized, collectivized Google-for-ideas. To knit these aims together, the W3C was working on a new version of HTML, called XHTML2, which would reinvent many of the Web page’s features with slightly better technologies, along with other standards. But, where W3C wanted to build a more open and accessible Web, the Web *industry* wanted to make the monkey dance.

It must be said that, in pursuit of its lofty goals, the W3C had become slightly unmoored. For example, the W3C’s Emotion Markup Incubator Group (<http://www.w3.org/2005/Incubator/emotion/>) intended to make it possible to annotate anything with *affect*. As they put it:

EmotionML provides mechanisms to represent emotions in terms of scientifically valid descriptors: categories, dimensions, appraisals, and action tendencies.

And immediately following that:

Given the lack of agreement in the community, EmotionML does not provide a single vocabulary of emotion terms, but gives users a choice to select the most suitable emotion vocabulary in their annotations.

There's no other way to put it. During the standardization process, emotions—obviously—ran high. And what did they accomplish? Using EmotionML, this is how you would indicate a pleasure value of 0.5:

```
<emotion dimension-set="http://www.w3.org/TR/emotion-  
voc/xml#pad-dimensions"><dimension name="pleasure"  
value="0.5"/></emotion>
```

EmotionML was also good at expressing anger, anxiety, hurt, and contempt. If the standard had made it to recommendation, and then had been widely adopted, you might have tagged political blog posts by their level of outrage, or tracked Barack Obama's crankiness during a press conference. Or you could have entered the Web and read only happy thoughts.

It was not to be. In 2004, while the W3C was working through its feelings, a lot of things were happening in the Web world. Google went public. Apple was selling millions of songs through its Web-technology-powered iTunes store. Apple had also launched its own browser, Safari.

And, so, a group of engineers from Apple, Mozilla (which makes Firefox), and Opera (which makes the Opera browser) formed a splinter group, known as WHATWG, the Web Hypertext Application Technology Working Group. It was, it said,

“increasingly concerned about the W3C’s direction with XHTML, lack of interest in HTML and apparent disregard for the needs of real-world authors.”

WHATWG set out to do what the W3C used to do—define a new version of HTML, called HTML5, that pulled together and standardized the incremental improvements that were appearing in Web browsers. These combined technologies would allow the Web browser to behave like a fast, general-purpose computer, with smarter forms, better video and audio playback, an improved model for turning documents into code, and a general rationalization and documentation of the enormous, tangled Web. The browser would henceforth be a place for applications—an engine that could run software. An operating system unto itself.

By 2007, the W3C, deep in the weeds, accepted the WHATWG approach as the right one, and adopted HTML5 as its own. XHTML2’s charter expired, and the emotive Web never came to pass. WHATWG stayed independent, and the two organizations began an uneasy collaborative alliance that continues to this day.

Despite W3C’s acceptance of HTML5, there remained many questions about how and when HTML5 would come into being. CNET’s Stephen Shankland has diligently tracked this standardization process for years; in 2010, he documented (<http://www.cnet.com/news/growing-pains-afflict-html5-standardization/>) the anger expressed:

Some examples of language that’s cropped up this month on the W3C’s HTML Working Group mailing list: “childish,” “intolerable,” “ridiculous,” “shenanigans.”

Also in 2010, Steve Jobs wrote an open letter to the universe, titled “Thoughts on Flash (<https://www.apple.com/hotnews/thoughts-on-flash/>),” in which he proclaimed that HTML5-style Web technologies were the way forward, not Adobe’s proprietary Flash platform (which performs many of the same complex application-

style tasks as HTML5, but which did not, and does not, run on an iPhone). The letter was a big deal, because it meant that HTML5 had the unequivocal blessing of one of the technology industry's largest companies, not as a document-delivery format but as an application-development framework.

Now, after being pored over for about seven years, HTML5 has arrived at a place of finality—sort of. As Shankland reports, the rift remains (<http://www.cnet.com/news/html5-is-done-but-two-groups-still-wrestle-over-webs-future/>) between WHATWG and the W3C. WHATWG sees the Web as an evolving, living platform; the W3C exists to formally recommend certain things. In practice, this means that it has been taking WHATWG's documents and changing them in ways that the WHATWG people don't always like. WHATWG refers to these adjustments as “forks.”

Who's in the right? Frankly, it doesn't matter that much. As far as the civilian, non-standards community goes, this is hermeneutics. The W3C has declared HTML5 done and is moving on to HTML5.1, which will include all the features that didn't make the first cut. WHATWG continues to document the Web in all its glory while publicly rolling its eyes at the W3C. Tremendous flareups occur, then settle, then threaten to flare up again. Revisions are made. WHATWG would clearly like the W3C to stop meddling and forking, but the W3C has been shepherding the Web for twenty years. For now, these two organizations have an uneasy accord. We can call that progress, and, if necessary, declare victory.

Why? Because it works. Browsers are now orders of magnitude faster and more reliable than they used to be. Code writers no longer see as many days lost to issues of incompatibility, or weeks lost to making something that works in Firefox also work in Internet Explorer. The browsers compete to make the Web faster, but no one seeks to change its core. The old, document-driven Web remains for anyone who wants to set up a Web page. The new,

application-driven Web works across platforms. It works across phones. It is complex, and sometimes confusing, but it largely works.

So how does the W3C's HTML5 standard (<http://www.w3.org/TR/html5/>) look in its totality? It's long. Looooong. Looooooooooooong. The section on images is fourteen thousand words, and it is just one tiny section in a figure-laden work running almost five hundred and thirty thousand words long—over five times as long as the HTML4 specification (<http://www.w3.org/TR/REC-html40-971218/>) it recommended, in 1997, though still a little shorter than “Infinite Jest.”

And yet, this massive, exhaustive spec describes barely a fraction of what truly defines the Web. For example, it doesn't say how one might go about making a JPG or a GIF, or how those files are arranged in binary streams of data. It simply tells you how to point to the images in a Web page. For all the other stuff, there are other standards, written in other rooms, by other people. And those standards are based on even more standards, all the way down, going back decades.

The Web started out as a way to publish and share documents. It is now an operating system: a big, digital sensory apparatus that can tell you about your phone's battery life, record and transmit your voice, manage your e-mail and your chats, and give you games to play. It can do this all at once, and with far less grand of a design than you might assume. That's the software industry: it promises you an Ellsworth Kelly, but it delivers a Jackson Pollock.

Unlike Microsoft's Windows or most of Apple's MacOS, the HTML5 standard is open and freely available to all. If you were to decide tomorrow to sit down and write your own Web browser, having become fed up with Chrome, Firefox, Safari, or Internet Explorer, you'd have every bit of information necessary to pull it off. Not one piece of knowledge would be denied you. There are

even tutorials to help get you started
(<http://limpet.net/mbrubeck/2014/08/08/toy-layout-engine-1.html>).

A standard is a skewed mirror of culture, and HTML5 is no different. Here is what it tells us we care about: words, headlines, video, and audio. We like to organize things into lists, and we like to look at pictures. And we want everything to be capable of animation and interaction—every letter, every tag, every structural element. Every bit of HTML5 is open to interpretation by code, available to be twisted, rotated, and manipulated by its users.

The Web, which used to be a place you went to get things, is now also a place to do things. That took a decade. It is 2014, and we have HTML5—the markup we deserve, and here to stay. Just like with the Buffalo Conference, of 1908, you will be able to hear the music a hundred years from now, as long as you have the right kind of player piano.

**This post was changed to more accurately describe the development of web apps, and to update Stephen R. Walli's job title.*

Paul Ford is a writer and programmer who lives in Brooklyn. He is writing a book of essays about Web pages for Farrar, Straus and Giroux.
