sitepoint

Article

The Importance of Web Standards

By Denis Boudreau
January 31st 2002
Reader Rating: 8.7

Denis Boudreau

Denis is an experienced Webmaster and technical writer. Aside from his day job as a Senior Web developer, he runs CYBERcodeur.net/Visceroide Design Multimedia, a Web company located in Montreal, Canada.

Back in early '96, the Web was merely more than a long list of text documents linked together by <a> tags, displaying an occasional image from time to time to spice things up. It had already come a long way from Tim Berners Lee's vision, but much was still to envision before it could achieve its full potential. In those days of old, needless to say, there wasn't much of a need for standardization as we mean it today.

Browsers were so forgiving in their interpretation of HTML [1] that just about anyone could master the basics and build a Web presence. The courageous few that used JavaScript [2] did so sparingly, in order to perform very basic tasks. Technologies such as DHTML [3] and Flash [4] were nowhere to be seen. Netscape Communicator 2.0 was in full swing and just about no one trusted Microsoft with its Internet Explorer browser. In this long-forgotten past, the World Wide Web was the exclusive playground of geeks and geniuses.

Boy... those were the days.

As the Web grew, so did the visions of thousands of developers worldwide. Recommendations started emerging from the W3C [5], in response to the pressure initiated by a leading-edge minority of Web builders who simply wanted more freedom in the expression of their work. However, most of the real pressure came from powerful browser companies who simply felt that nobody -- including the people who actually built the Web -- had the right to tell them how they should or should not act.

These companies took it upon themselves to support new markup tags that would only work in their particular product. However, some of these tags were so bad that they only helped broaden the chasm of incompatibility between browsers. This continued to the point where building a Website that would display equally well in all versions of all browsers was almost impossible.

HTML 4.0 and CSS [6]-1 were brought forth as official Web standards from the W3C in an attempt to put an end to the chaos. For the first time in history, developers were given means to standardize code by separating the content of HTML documents from the presentation layout. Ingenious workarounds and hacks, while very effective and clever, led to the saturation of HTML. The language had become anything but what it had originally been intended as - a metalanguage that would add structure to text documents. It was time to turn to better, more able tools.

This need saw the advent of the promising XML [7] and DOM [8]. HTML 4.0 evolved into XHTML [9] 1.0 and DHTML matured from its shaky debuts to an efficient state of reliability. New browser versions were introduced that supported the standards even more completely. Thus the future of the Web was bright once again...

The relevance of Web standards is most obvious when we consider emerging technologies. In these times of tremendous growth, the Web needs guidance in order to reach its full potential - and standards can serve as the perfect guides to help realize that potential. Let's now take a look at the people behind the standards, consider the standards themselves, and touch on some of the reasons why you should comply with them.

Who's Behind the Web Standards?

Everybody's heard about Web standards, yet very few really go the extra mille to comply with them. Currently, five standards have been officially implemented, and two more are in the works. Most of them come from the W3 Consortium. One standard, however, originated at another organization, called ECMA.

The W3C develops open specifications to enhance the interoperability of Web-related products, in an effort to ultimately standardize the WWW. Special working groups obtain a general consensus from companies and organizations that are active in the Web's development. These documents, called Working Drafts and Proposed Recommendations, are submitted to the W3C board of executive members and directors for formal approval as "official" W3C Recommendations. These Recommendations are what the Web standards are all about. They are the foundations upon which the future of Web building resides. They include HTML, XML, CSS and DOM.

ECMA's main role is to develop standards and technical reports in the field of information and communication technology. The Web standard brought up by ECMA has been accepted as a base for scripting standards worldwide. It's called ECMAScript, or ECMA-262, and it's a standardized, cleaner version of JavaScript.

What are the Standards?

- **HTML (HyperText Markup Language)**, the first ever standard to emerge from the W3C, is widely used on the Web. It is by far the most common tool for designing Web pages, which is why people usually refer to HTML as the backbone of the WWW. Thanks to HTML structuring elements (tags), text documents interpreted by browsers can display their contents according to specified formats. HTML is one of the main reasons why the Web has become so popular over the years.
- **XML (eXtensible Markup Language)** is like HTML on steroids. While it supports most of the fixed set of elements contained in HMTL 4.0, it allows the coder to customize new tag elements according either to a pre-set DTD (Document Type Definition) or to one that can be personally defined. XML is much more flexible than HMTL and is totally future-oriented. XML is the standard through which the Web's full potential can be achieved.
- **CSS (Cascading Style Sheets)** are a Web developer's dreams come true. It is a mechanism through which changes in appearance and position can be assigned to HTML or XML elements, simply by declaring specific styles to them. The overall appearance of entire sites can be so defined with the use of CSS, allowing it to be remodeled in a matter of seconds. This Standard was brought forward by the W3C to create a simpler and more structured World Wide Web.

- **DOM (Document Object Model)** allows a scripting language to exert its full power and interactive capabilities on a Web page. It's a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of Web documents. They can be further processed, and that processing result can then be incorporated back into the Web page by the browser. The use of DOM as a standard will simplify DHTML coding, and will be fully compatible with upcoming technology improvements.

- **ECMAScript (the standardized version of JavaScript)** is a client-side, object-oriented [10] scripting language based on several technologies such as Netscape's JavaScript and Microsoft's Jscript. The main use of ECMAScript is to manipulate objects in a Web document that are specified by the DOM. Through it, elements can be manipulated, moved, or have some properties changed, allowing Web developers to implement such effects as animated text and graphic rollovers. The current specification is in its third edition, as ECMA-262.

The W3C is also working on two recommendations to be introduced as standards in the near future: these are XHTML and XSL [11]. XHTML (eXtensible HyperText Markup Language) is a reformulation of HTML 4.0 into an XML application. XSL (eXtensible Stylesheet Language) will be a mechanism though which XML documents will be transformed and displayed using a vocabulary specifically designed for XML.

So that's it folks: those are the standards upon which the Web should eventually rest. Others will inevitably crop up and some are bound to evolve or maybe even be dropped altogether... But while this is all very nice, one question still remains: why should anyone care at all?

Standards - Why Bother?

The use of standards automatically makes every page you build genuinely cross-browser and cross-platform. Anyone who's been active in Web development for more than 15 minutes knows how time-consuming and complicated it is to double-code for different browser versions. Standards help you avoid such tedium.

1. Achieve a More Stable Web

With the arrival of new hardware like mobile phones and hand-held devices, coding 100% compatible Web applications will only get harder, if not simply impossible. Following the standards is the most efficient way to achieve a better, more stable Web, where applications we can only dream of today can be turned into the reality of tomorrow.

2. Reduce Development and Maintenance Time

Coding in accordance with standards can shorten both development and maintenance time. Debugging and troubleshooting therefore becomes easier, because the code follows a pattern. Accessibility [12] is another very important issue behind the Web standards: not only does it mean allowing the Web to be fully used by people with disabilities, it also means allowing people with unconventional browsers to have access to any Web page.

3. Allow Backward Compatibility and Validation

Standards are written to be compliant with older browser versions. This is what the W3C refers to as "backward compatibility". While these browsers may not show everything as initially intended, they will still understand the basic structure of Web documents, and display them accordingly. Compliant code can also be validated through a validation service. This, again, makes the developer's work a lot easier and can save a lot of production time.

4. Increase Search Engine Success

From a page ranking point of view, complying with standards also guarantees better results and increased visibility in search engines. Standards-compliant documents get indexed more accurately due to the structural information present in them, as they can be easily accessed and evaluated by mechanical and human methods alike. Compliant code can also be easily converted to other formats, such as databases or Word documents. This allows for more versatile use of the information within the document. It also allows a simplified migration to new systems such as televisions and PDAs.

5. Graceful Degradation Now, and in the Future

Another important factor to remember is that most Web standards are designed with forward and backward compatibility in mind. This means that Web pages coded for old versions of the standards will still be supported in new versions of browsers, while Web pages coded for new versions of the standards will gracefully degrade to produce an acceptable result in older browsers.

6. Common Knowledge

Web standards offer a set of rules that every developer can follow, understand, and become familiar with. When one developer builds a site according to standards, another will be able to pick up where the former left off as easily as if he had been the one designing it in the first place.

Wrapping Up

The Web has come a long way since the creation of the W3C in 1994. While the Consortium actively works at bettering things, much is still to achieve and it is up to Web developers worldwide to make the dream of a better Web a reality. The reasons mentioned above are some of the most important reasons why Web standards should matter to everyone.

Who knows how many people will access the Web on a cell phone, even next year? Is it 10% of Internet users? 20%? More even? Who knows? The only thing we can predict for sure is that a percentage of potential customers are likely to say particular ecommerce sites suck big time because they can't be accessed from a specific Internet output device or older browser version. This is therefore an equivalent percentage of potential online sales that will go up in smoke for those retailers who don't comply to the standards. Users are never wrong. You will always be the one identified as a bad developer.

Now that oughta hurt, don't you think?

A talented Web developer will always be a talented Web developer whether he has to hack around his code or not. It seems much more appealing to be recognized as a talented standards application specialist oriented towards the

Web's future than a talented workaround code-hacker specialized in technologies that are becoming more obsolete by the day.

Back to SitePoint.com

[1] /glossary.php?q=H#term_75
[2] /glossary.php?q=J#term_9
[3] /glossary.php?q=D#term_7
[4] /glossary.php?q=F#term_16
[5] /glossary.php?q=W#term_49
[6] /glossary.php?q=C#term_8
[7] /glossary.php?q=X#term_3
[8] /glossary.php?q=D#term_39
[9] /glossary.php?q=X#term_63
[10] /glossary.php?q=O#term_10
[11] /glossary.php?q=X#term_32
[12] /glossary.php?q=A#term_61

» Page 1