



[The Web Standards Project](#) is a grassroots coalition fighting for [standards](#) that ensure simple, affordable access to web technologies for all.

- [home](#)
- [about](#)
 - [bios](#)
 - [colophon](#)
 - [contact](#)
 - [history](#)
 - [mission](#)
 - [sitemap](#)
- [act!](#)
- [learn](#)
- [opinion](#)
- [press](#)

WaSP: Fighting for Standards

The World Wide Web Consortium ([W3C](#)), along with other groups and standards bodies, has established technologies for creating and interpreting web-based content. These technologies, which we call “web standards,” are carefully designed to deliver the greatest benefits to the greatest number of web users while ensuring the long-term viability of any document published on the Web. Please see the sidebar for details.

Designing and building with these standards simplifies and lowers the cost of production, while delivering sites that are accessible to more people and more types of Internet devices. Sites developed along these lines will continue to function correctly as traditional desktop browsers evolve, and as new Internet devices come to market.

It sounds so straightforward and makes so much sense. So what’s the problem? And why is there a Web Standards Project?

The Problem

Though leading browser makers have been involved in the creation of web standards since W3C was formed, for many years compliance was observed in the breach. By releasing browsers that failed to uniformly support standards, manufacturers needlessly fragmented the Web, injuring designers, developers, users, and businesses alike.

Lack of uniform support for W3C standards left consumers frustrated: when using the “wrong” browser, many could not view content or perform desired transactions. Among those most frequently hurt were people with disabilities or special needs.

Quandaries and Costs

At the same time, lack of uniform support for W3C standards left designers, developers and site owners in a terrible quandary: could they afford to implement multiple versions of every web page in order to accommodate incompatible browsers? If not, which browsers should they neglect, and how many millions of potential visitors were they willing to turn away? Either way, the cost was too high. It still is.

The fractured browser market added at least 25% to the cost of developing all sites. Through lack of budget, many developers produced sites that locked out potential customers. Many developers who knew about standards saw no point in developing sites for browsers that did not support them. Others knew little or nothing about standards—and many still don't, including those at multi-million-dollar agencies who seem to grasp ASP, Java, Flash MX, and .Net, yet understand almost nothing about structural markup, style sheets, and the importance of separating structure from presentation.

Some designers, stymied by browser incompatibility, deliberately excluded all but the oldest and most universal web technologies from their sites. Such sites often succeeded at functioning in all browsers, but at the cost of limited consumer appeal and functionality.

Others relied on visual editors and publishing tools to generate multiple layers of markup and code optimized for the quirks of various popular browsers. This wasted money as well as bandwidth, and often generated sites that stopped working in the next generation of browsers (and never worked at all in alternative browsers or devices, from screen readers to Lynx to PDAs to less popular browsers such as Opera). The Web is littered with the corpses of once-impressive sites that cannot function in contemporary browsers or devices. Making matters worse, such sites are still being created every day.

Some designers became so frustrated they turned their backs on web standards altogether, and began to develop exclusively in proprietary environments. Though rich in creative potential, such technologies suffer from lack of broad accessibility, and fail to provide for common needs such as bookmarking, printing, copying and pasting, and other tasks web users must perform on informational and transactional sites.

Born of Necessity

In response to these problems, The Web Standards Project (WaSP) was formed in 1998 with the goal of promoting core web standards and encouraging browser makers to do the same, thereby ensuring simple, affordable access for all.

Though our message initially met with resistance (particularly from browser companies' marketing and P.R. departments), eventually we prevailed—in part because engineers at many browser companies agreed with us and saw WaSP as an ally in their internal battles with management.

Beginning in 2000, one leading browser after another delivered on the promise of many of the standards we had (sometimes shrilly) promoted. Current market-leading browsers, along with several of their competitors, provide excellent support for HTML 4, XHTML 1, CSS, ECMAScript (the standard version of JavaScript), and the DOM—or are on the road to such compliance.

Thanks to these browsers, designers and developers are finally free to build with XHTML and CSS, and in many cases can separate structure from presentation to maximize portability and accessibility. With care, designers and developers may also use the W3C standard DOM to add sophisticated behavior to their sites.

So what's the problem, and why is there still a Web Standards Project?

Challenges Ahead

Though today's browsers support standards, tens of thousands of professional designers and developers continue to use outdated methods that yoke structure to presentation, in some cases entirely avoiding semantic structures and misusing (X)HTML as a design tool. Highly paid professionals continue to churn out invalid, inaccessible sites filled with structurally meaningless markup, huge image maps, excessively nested tables, and outdated detection scripts that cause the very usability problems they were originally intended to prevent.

Many books on web development still teach outdated methods, and many practitioners take pride in delivering sites that look and work exactly the same in compliant and non-compliant browsers alike, at the cost of accessibility, long-term viability, or forward compatibility. Others develop proprietary code that works only in a handful of popular browsers.

Thus one of WaSP's primary goals is to provide educational resources that can help our peers learn standards-compliant methods that are in their interest and that of their clients and site users.

Many professionals accomplish their work by means of visual editing environments developed at the height of the Browser Wars. As mentioned above, such tools by default create invalid, non-semantic sites optimized for 4.0 browser quirks instead of standards. In 2002, two leading visual editors vastly improved their support for web standards and accessibility (one of them with help from The Web

Standards Project). But to make use of these improvements, professionals must learn the basics and benefits of designing and building with web standards. This again points to the need for developer education.

Clients and site managers also need this information if they seek to create sites that are accessible in today's browsers and devices and will remain viable as browsers and devices evolve. It is WaSP's hope that, once informed of the benefits standards provide, site owners will stop viewing their sites as a species of print advertising that must look exactly the same in all environments. And that they will focus instead on delivering appropriate content and functionality within the context of presentations that may vary slightly according to the needs and capabilities of differing browsers and devices.

Original Mission Statement

The WaSP's original [1998 Mission Statement](#) is available at archive.webstandards.org.

© 1998-2002 Web Standards Project, www.webstandards.org

Legal

- [German \(de\)](#)
- [English \(en\)](#)
- [French \(fr\)](#)
- [Italian \(it\)](#)
- [Japanese \(ja\)](#)
- [Dutch \(nl\)](#)
- [Portuguese \(pt\)](#)

What “Standards?”

When we speak about “standards” for the Web, we mean:

Structural Languages

[Extensible Hypertext Markup Language \(XHTML\) 1.0](#)

[XHTML 1.1](#)

[Extensible Markup Language \(XML\) 1.0](#)

Presentation Languages

[Cascading Style Sheets \(CSS\) Level 1](#)

[CSS Level 2](#)

[CSS Level 3](#)

Object Models

[Document Object Model \(DOM\) Level 1 \(Core\)](#)

[DOM Level 2](#)

Scripting Languages

[ECMAScript 262](#) (the standard version of JavaScript)

Additional Presentation Languages (Markup)

[Mathematical Markup Language \(MathML\) 1.01](#)

[MathML 2.0](#)

[Scalable Vector Graphics \(SVG\) 1.0](#)

... as well as emerging standards, such as those for television- and PDA-based browsers.