

Working together for standards

The Web Standards Project

Frequently Asked Questions (FAQ)

What are web standards and why should I use them?

1 [Introduction](#)

2 [What are the standards?](#)

2.1 [W3C Standards](#)

2.1.1 [What is the W3C](#)

2.1.2 [What does it do?](#)

2.1.3 [What are the W3C standards?](#)

2.1.3.1 [HTML 4.0](#)

2.1.3.2 [XML 1.0](#)

2.1.3.3 [XHTML 1.0, 1.1, XHTML Modularization](#)

2.1.3.4 [CSS - Cascading Stylesheets](#)

2.1.3.5 [DOM](#)

2.2 [ECMA Standards](#)

2.2.1 [What is the ECMA?](#)

2.2.2 [What does it do?](#)

2.2.3 [What are the ECMA standards?](#)

2.2.3.1 [ECMAScript](#)

3 [What are the advantages of using Standards?](#)

3.1 [Accessibility](#)

3.1.1 [To software/machines](#)

3.1.2 [To people](#)

3.2 [Stability](#)

4 [Conclusions](#)

1. Introduction

Tim Berners-Lee's dream for his invention, the World Wide Web, is a common space where users can share information to work together, to play, and to socialize ([The World Wide Web, A very short personal history](#)). As web developers, creating business, social, and educational sites, we turn this dream into reality.

But in this period of tremendous growth, the Web needs guidance to realize its full potential. Web standards are this guidance. These standards help ensure that everyone has access to the information we are providing, and also make web development faster and more enjoyable.

Standards compliance makes it easier for people with special needs to use the Web. Blind people may have their computer read web pages to them. People with poor eyesight may have pages rearranged and magnified for easier reading. And people using hand-held devices can browse the Web just as easily as those using high-end workstations.

As we will explain, there are also many [practical reasons](#) for developers to be concerned with web standards. Search engines can do a better job of indexing sites, for example. Using browser-specific code often doubles or triples the work to create web pages, and leaves a lot to be desired when new media are introduced. This situation will only get worse without the sound direction of web standards.

Some people fear that standards are limiting. In reality, they remove much of the tedious labour involved in web development, and give developers more time and more flexibility to be truly creative. They are both open to future improvement and mindful of past technology.

Many uses of the Web, including some that are only dreamed of today, will not be possible, or will be more difficult, without widespread standards compliance. At the moment, there are systems and software that are very common, seemingly close

to universal, but who knows what tomorrow will bring? Tying ourselves to the control of any single company means limiting our future to the fortunes and misfortunes which that one company can or will provide. Maintaining universal standards will allow the Web to survive while encouraging innovation to continue at its current pace.

Standards have so much to offer that we at [The Web Standards Project](#) (WaSP) consider it necessary to help you learn more about them. This document is merely a starting point; it will give you a solid understanding of what standards exist, why they do, and why you should care about them. Every time we create a piece of the Web, we contribute to the common information space that is the Web. We can build it up, or we can add weight that will tear it apart. The choice belongs to us; the consequences belong to everyone.

2 What are the standards?

2.1 W3C Standards

2.1.1 What is the W3C?

The [World Wide Web Consortium](#) (W3C) is an international industry consortium dedicated to "[leading the Web to its full potential](#)". It's led by Tim Berners-Lee, the inventor of the Web. Founded in 1994, the W3C has more than 450 [member organizations](#) - including [Microsoft](#), [America Online](#) (parent company of [Netscape Communications](#)), [Apple Computer](#), [Adobe](#), [Macromedia](#), [Sun Microsystems](#), and a variety of other hardware and software manufacturers, content providers, academic institutions, and telecommunications companies. The Consortium is hosted by three research institutions - [MIT](#) in the US, [INRIA](#) in Europe, and [Keio University](#) in Japan.

2.1.2 What does it do?

The W3C develops open specifications (de facto standards) to enhance the interoperability of web-related products. W3C Recommendations are developed by working groups consisting of Consortium members and invited experts. Working groups obtain general consensus from companies and other organizations involved in creating applications for the Web, and create Working Drafts and Proposed Recommendations. These are then submitted to the W3C membership and director, for formal approval as W3C Recommendations. More information regarding this process and the review stages can be obtained from the [W3C website](#).

2.1.3 What are the W3C standards?

2.1.3.1 HTML 4.0 - HyperText Markup Language

HyperText Markup Language (HTML) is widely used on the Web for adding structure to text documents. Browsers interpret these documents, representing the structure in media-specific ways to the user. For example, visual browsers typically display the `` element (` ... `) as bold text, while text-to-speech readers might emphasize that text when pronouncing it.

With the help of [Cascading Style Sheets](#) (CSS) the author may define how structural elements are to be represented, overriding the browser defaults.

2.1.3.2 XML 1.0 - Extensible Markup Language

Extensible Markup Language (XML) is a markup language like [HTML](#), but instead of having a single, fixed set of elements, it allows you to define your own - or use a set made by someone else. It even allows using multiple sets within a single document - by using [XML namespaces](#).

Some applications of XML, such as [XHTML](#) and [MathML](#), have already become W3C Recommendations. Others are currently W3C Working Drafts.

Style sheet standards, such as CSS and XSL, offer a variety of options for specifying how XML elements are to be rendered. Standards-compliant support for direct rendering of XML is spotty in browsers, so for presenting information to humans, HTML (or XHTML) with CSS-driven styling is the way to go. XML is mostly used for machine-to-machine communication today.

XML is more flexible than HTML, primarily because of

Example of part of an XML document

```
<addressbook>
  <entry>
    <name>Bill Gates</name>

    <email>bgates@microsoft.com</email>
  </entry>
  <entry>
    <name>Marc Andreessen</name>
    <email>marca@netscape.com</email>
  </entry>
  <entry>
    <name>Jon S. von Tetzchner</name>
    <email>jon@opera.com</email>
  </entry>
```

the ability to add your own elements and make your own structural systems. This makes it an ideal ~~format for the~~ format for the organization of large quantities of data - it is already in use in many databases and search engines.

2.1.3.3 XHTML 1.0, 1.1, and Modularization

XHTML 1.0 is a reformulation of HTML as an XML application. XHTML 1.0 can be seen as ideologically coming from HTML 4.01, and being technically stricter because of XML's influence.

XHTML will display in your browser identically to the equivalent HTML. You might want to use XHTML if there is any chance you're going to need to reprocess your content, for example to send it to a PDA; XML's stricter syntax rules make automatic processing of XHTML much easier and cheaper than ordinary HTML.

Ideologically, XHTML 1.0 inherits the following general concepts from HTML 4.01:

- That presentation and document formatting should be separated via style sheets
- That documents should be made accessible
- That documents should be internationalized

XHTML 1.0 also uses the model of three DTDs: Strict, Transitional, and Frameset. This model originally emerged in HTML 4.0 and followed through to HTML 4.01.

Some important technical practices from XML onto XHTML includes:

- That all document types are declared via the correct DOCTYPE declaration
- That the structure of a conforming document contain the DOCTYPE declaration, an html element with the XHTML namespace declared, a head element including the title element, and a body element
- That all elements and attribute names are written in lower case, and that all attribute values are quoted
- That all non-empty elements (e.g. p, li) are properly terminated with a closing tag
- That all empty elements (e.g. br, hr, img) are properly terminated with a trailing slash (
)
- That documents validate against the DTD that is declared

For templates, please see [Learn > Templates](#)

XHTML 1.1 is made up of three primary parts:

- The XHTML 1.0 Strict DTD (with minor modifications)
- XHTML Modularization
- The Ruby Annotation

If you'd like to author documents in XHTML 1.1, you can do so in a couple of ways. The first is by using the public XHTML 1.1 DTD. By doing this, your work will be extremely structured because there are virtually no presentational attributes in XHTML 1.1. The separation of structure and presentation is complete here, and all of your presentation work will go in a style sheet.

Another means of authoring documents in XHTML 1.1 is to tap into XHTML Modularization. This is the breakdown of familiar components of HTML and XHTML (such as text, tables, frames, forms) into discrete chunks. You can then write your own DTD and use only those components you require. This is extensibility in action, essentially giving you, the web author, the opportunity to customize your markup.

The Ruby Annotation is a special means of dealing with certain Asian character annotations. Ruby falls under the work being done with Internationalization.

2.1.3.4 CSS - Cascading Style Sheets

Cascading Style Sheets (CSS) is a mechanism for changing the appearance of [HTML](#) or [XML](#) elements, by assigning styles to element types, self-defined classes of elements or individual instances.

Stylesheets can be used to consistently define the appearance of an entire site. Following the introduction of CSS, the W3C recommended that layout-specific features in HTML be phased out and replaced by stylesheets, creating a simpler and more structural World Wide Web.

2.1.3.5 DOM 1 - Document Object Model Level 1

The DOM allows the full power and interactivity of a scripting language (such as [ECMAScript](#), the standardized version of

JavaScript) to be exerted on a web page. (In programming terms, the Document Object Model (DOM) Level 1 is an Application Programming Interface (API) for interacting with web pages.) It gives the scripting language easy access to the structure, content, and presentation of a document which is written in such languages as [HTML](#) and [CSS](#).

The DOM is compatible with future improvements in technology; it will allow any scripting language to interact with whatever languages are being used in the document. This standard will not only make it easier to program dynamic HTML, but will also make adapting to future Internet technology much less painful.

2.2 ECMA Standards

2.2.1 What is the ECMA?

The [European Computer Manufacturers Association](#) (ECMA) is an organization officially founded in 1961 in order to meet the need for standardizing computer operational formats, including programming languages and input/output codes.

The ECMA is based in Geneva, Switzerland, near the headquarters of the [International Organization for Standardization](#) (ISO) and the [International Electrotechnical Commission](#) (IEC). In 1994, the organization's name was changed to the ECMA - European Association for Standardizing Information and Communication Systems, in order to reflect its broader range of activities.

2.2.2 What does it do?

The main role of the ECMA is to develop Standards and Technical Reports in the area of information and communication technology. As ECMA is an association of companies and not an official standardization institute, they often collaborate with official national or international institutes.

ECMA Standards have been accepted as a base for international and European standards. So far more than [270 ECMA Standards and 70 Technical Reports](#) have been published.

Of these standards 85 have been accepted as international standards by the [International Organization for Standardization](#) (ISO). In addition, 25 have been accepted as European standards by the [European Telecommunications Standards Institute](#) (ETSI).

2.2.3 What are the ECMA standards?

2.2.3.1 ECMAScript (standardized JavaScript)

ECMAScript is a standardized scripting language, based largely on [Netscape's JavaScript](#) and [Microsoft's JScript](#). The ECMAScript standard is defined by ECMA's [Technical Committee 39](#) (TC-39).

The main use of ECMAScript, which is an object-based language, is to manipulate the objects in web pages which are specified by the [Document Object Model](#) (DOM). These objects (effectively, the elements which make up web pages, or the web pages as a wholes) can then be added to, deleted, moved, or have their properties changed. This lets web developers implement such effects as animated text, graphic roll-overs, and pages that change based on user input without having to be reloaded.

The current ECMAScript specification is ECMA Standard ECMA-262, [ECMAScript Language Specification, 2nd edition](#).

3 What are the advantages of using web standards?

3.1 Accessibility

3.1.1 To software/machines

Complying with web standards can give your web pages greater visibility in web searches. The structural information present in compliant documents makes it easy for search engines to access and evaluate the information in those documents, and they get indexed more accurately.

Because use of web standards makes it easier for server-side as well as client-side software to understand the structure of your document, adding a search engine to your own site becomes easier and gives better results.

Standards are written so that old browsers will still understand the basic structure of your documents. Even if they can't understand the newest and coolest additions to the standards, they'll be able to display the content of your site. The same, of course, applies to robots - systems that collect information from your site on behalf of search engines and other indexers.

Compliant code gives you the opportunity of validating your page with a validation service. Validators process your

documents and present you with a list of errors. This makes finding and correcting errors a lot easier, and can save you a lot of time.

Compliant documents can easily be converted to other formats, such as databases or Word documents. This allows for more versatile use of the information within documents on the World Wide Web, and simplified migration to new systems - hardware as well as software - including devices such as TVs and PDAs.

3.1.2 To people

[Accessibility](#) is an important idea behind many web standards, especially HTML.

Not only does this mean allowing the web to be used by people with disabilities, but also allowing web pages to be understood by people using browsers other than the usual ones - including voice browsers that read web pages aloud to people with sight impairments, Braille browsers that translate text into Braille, hand-held browsers with very little monitor space, teletext displays, and other unusual output devices.

As the variety of web access methods increases, adjusting or duplicating websites to satisfy all needs will become increasingly difficult (indeed, some say it's impossible even today). Following standards is a major step towards solving this problem. Making your sites standards-compliant will help ensure not only that traditional browsers, old and new, will all be able to present sites properly, but also that they will work with unusual browsers and media.

Some consequences of ignoring standards are obvious: the most basic consequence is that you will restrict access to your site. How much business sense does it make to limit your audience to only a fraction of those who wish be a part of it? For a business site, denying access to even small portions of a target audience can make a big difference to your profit margin. For an educational site, it makes sense to allow access not only to affluent, able-bodied school-children with graphical browsers, but also to children in regions with poorly-developed infrastructure who are best served by text-only browsing, or disabled students using specialized browsers.

The same principle applies to all types of websites — while straying from the standards and taking advantage of browser-specific features may be tempting, the increased accessibility which comes from standards-compliance will lead to far greater rewards in the long run.

3.2 Stability

Most web standards are generally designed with forward- and backward-compatibility in mind — so that data using old versions of the standards will continue to work in new browsers, and data using new versions of the standards will “gracefully degrade” to produce an acceptable result in older browsers.

Because a website may go through several teams of designers during its lifetime, it is important that those people are able to comprehend the code and to edit it easily. Web standards offer a set of rules that every Web developer can follow, understand, and become familiar with: When one developer designs a site to the standards, another will be able to pick up where the former left off.

4. Conclusion

As web developers, we are constantly trying to address the problem of inconsistencies between the renderings of web pages by different browsers and browser versions. This necessitates either time-consuming double/multiple coding, or coding for a single browser which makes it harder, if not impossible, for some of the public to use the site. This situation will be made even worse with the advent of additional hardware and software which will be able to browse the Web, such as telephones, pagers, and PDAs.

Web standards are not arcane laws decreed by ivory-tower organizations. [As we have described](#), the standards are for the most part decided by representatives of the same people who use them — browser makers, web developers, content providers, and other organizations.

Writing web pages in accordance with the standards shortens site development time and makes pages easier to maintain. Debugging and troubleshooting become easier, because the code follows a standard. No longer do you have to worry about the coding and maintenance for several versions of code that are supposed to accomplish the same presentation. *One* version of your site, and that is it.

The universal adoption of web standards is becoming of paramount importance. The mission of [The Web Standards Project](#)

is to make the Web a better place, for developers and for end-users, by encouraging browser and web page editor makers to follow the standards in their applications. This effort will be greatly helped when web developers *use* the standards as a matter of course, and insist that generators and renderers of their code comply with the standards.

The [reasons we have given](#) should give you, the web developer, plenty of incentive to begin using standards, and also plenty of ammunition with which you can encourage your place of business and fellow developers to use those standards.

[Help make the dream a reality.](#)

Web Standards Project Developer Education Committee: *Stephan Nedregard (coordinator), Kynn Bartlett, Gail T. Cohen, Jens Edlund, Nick Finck, Tomas Fjetland, Peter Fleck, Markus Gut, Holger Maier, Julian Missig, Laura Mollett, Randy Piatt, Lewis A. Shadoff, Juergen Steinwender, Bart Szyszka, Matthew Thomas, Dane Weber*

Updated 02-27-2002 by Molly E. Holzschlag and Shirley E. Kaiser of WaSP LEARN Committee.

Translations:

[FAQ \[French Translation\]](#)

[FAQ \[Japanese Translation\]](#)

[FAQ \[Italian Translation\]](#)

[FAQ \[Dutch Translation\]](#)

[FAQ \[Simplified Chinese Translation\]](#)

[FAQ \[Traditional Chinese Translation\]](#)

[FAQ \[Polish Translation\]](#)

[FAQ \[Korean Translation\]](#)

All of the entries posted in WaSP Buzz express the opinions of their individual authors. They do not necessarily reflect the plans or positions of the Web Standards Project as a group.

This site is valid [XHTML 1.0 Strict](#), [CSS](#) | Get Buzz via [RSS](#) or [Atom](#) | [Colophon](#) | [Legal](#)