Protocols

**Protocol**, in computer science, a set of rules or procedures for transmitting data between electronic devices, such as computers. In order for computers to exchange information, there must be a preexisting agreement as to how the information will be structured and how each side will send and receive it.

A **network protocol** is an established set of rules that determine how data is transmitted between different devices in the same **network**. Essentially, it allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure or design.

**Below are the different types of Networking Protocols.**

* **HTTP** or HTTPs. This stands for **Hypertext Transfer Protocol** or **Hypertext Transfer Protocol** (secure). ...
* **FTP (File Transfer Protocol**) ...
* **Email Protocols** (**POP3**, **IMAP**, **SMTP**) ...
* TCP(**Transmission Control Protocol**) and **UDP**(**User Datagram Protocol**)

**HTTP** is unsecured while **HTTPS** is secured. **HTTP** sends data over port 80 while **HTTPS** uses port 443. **HTTP** operates at application layer, while **HTTPS** operates at transport layer. No SSL certificates are required for **HTTP**, with **HTTPS** it is required that you have an SSL certificate and it is signed by a CA

**Web Standards: The What, The Why, And The How**

Web Standards, and the documentation to support them, provide huge insight into ‘the why’ and ‘the what’ of the world wide web. They are a fantastic resource for any web developer and help people build stuff for the web that is functional, accessible and cross-compatible. In this article, we take a look at the history of Web Standards, how to use them in your work and ways you can get involved in making them.

The World Wide Web is an interesting place.

As the Internet has grown and become more common place, it has become a gigantic instrument of change in terms of the way in which we interact with the world and each other.

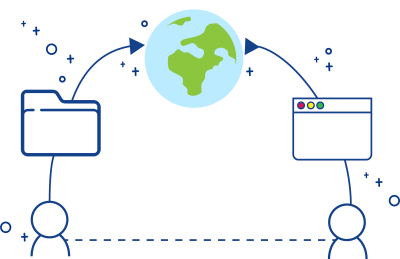
Like many people, my intro to web development at school was kind of bleak. Our school ICT (Information Computing Technology) lessons taught us very little, using Dreamweaver (back when it was a Macromedia product) as a platform to visually edit a personal website with the biggest lesson being “what is a hyperlink”. We didn’t even view the HTML source of our own websites!

* An introduction to web standards (what are they, why do they exist and who makes them);
* How to navigate and make use of standards in your work;
* Ways you can get involved in contributing to new and existing standards.

Let’s begin our introduction to web standards by asking, “*Why do we need standards for the web?*”

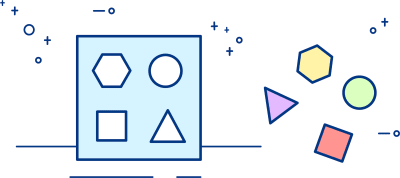
**The World Wide Web Before Standards**

We can think of the world wide web as an information ecosystem. People create content that is fed into the web. This content is then passed through a browser to allow people to access that information.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/4fcca3cf-937f-4c3d-a4a5-a914aa960445/web-standards-ecosystem.png)([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/4fcca3cf-937f-4c3d-a4a5-a914aa960445/web-standards-ecosystem.png))

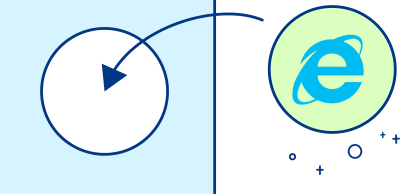
Before Web Standards, there weren’t many fixed rules for any part of this system; no formal rules as to how the content should be created, nor any requirements in terms of how a browser should serve up that information to the people that are requesting it.

So, in a way, the web operated a bit like that children’s toy where you have to sort the different shaped blocks into the correct holes. In this analogy, the different types of browsers are the different shaped holes and the content or websites, are the brightly colored blocks.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/9ab262ce-005e-4b2e-9cb9-783ee7ce02ab/web-standards-sorting-toy.png)The sorting shape toy and its colorful blocks ([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/9ab262ce-005e-4b2e-9cb9-783ee7ce02ab/web-standards-sorting-toy.png))

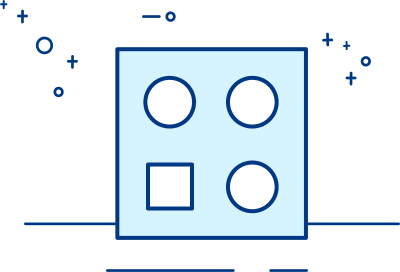
In the past, as a content creator you would make a website to fit the browser it would be intended for. For example, you would create an IE-shaped block to be able to pass this through the Internet Explorer hole.

This meant that this website block you had created would only fit through that one hole and you would need to rebuild your content into other shapes for it to be viewed using any of the other browsers.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a25247e1-6bf6-47d2-b452-7e42b813fe51/web-standards-internet-explorer-block.png)Fitting an IE-sized block into an IE-sized hole ([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a25247e1-6bf6-47d2-b452-7e42b813fe51/web-standards-internet-explorer-block.png))

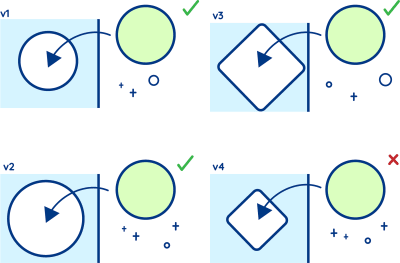
Developers in the 90s would often have to make three or four versions of every website they built, so that it would be compatible with each of the browsers available at the time. And what is more, browser makers in attempts to better their competition would introduce “features” that diversified their approach from their competitors.

In the beginning, it was probably fairer to say our Internet browser to content-matching toy looked more like this:

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/8297617f-28ac-4397-a7ef-823533c8acb0/web-standards-early-sorting-toy.png)A sorting toy with three round holes and one square hole ([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/8297617f-28ac-4397-a7ef-823533c8acb0/web-standards-early-sorting-toy.png))

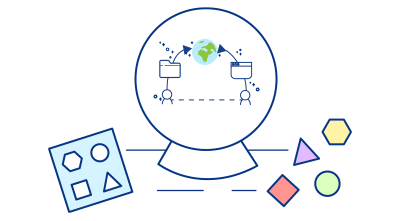
This was because browsers were built to handle pretty much the same stuff, which was largely text-based content. So, for the most part, a website block would fit through the majority of the holes, with the exception of maybe one where it might fit — but not perfectly.

As the browsers developed, they begin to add features (e.g. by changing their shape) and it became more and more difficult to make a block that would pass through each of the browser holes. This even meant that a block that could once fit through one particular hole, didn’t fit through that hole any longer; adding these features into the browser would often result in poor reverse compatibility.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a8083573-5cfa-4023-9c2a-8a2e20403ffc/web-standards-changing-holes.png)A hole that changes over time means all blocks will not always fit through. ([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a8083573-5cfa-4023-9c2a-8a2e20403ffc/web-standards-changing-holes.png))

This was really damaging for some developers. It created a system in which compatibility was limited to the content creators that could afford to continuously update and refactor their websites for each of the available browsers. For everyone else, every time a new feature or version was released, there was a chance your website would no longer work with that browser.

Web standards were introduced to protect the web ecosystem, to keep it open, free and accessible to all. Putting the web in a protective bubble and disbanding with the idea of having to build websites to suit specific browsers.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a3cf09ab-3643-4183-8d58-6f5e049e7823/web-standards-web-in-bubble.png)([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/a3cf09ab-3643-4183-8d58-6f5e049e7823/web-standards-web-in-bubble.png))

When standards were introduced, browser makers were encouraged to adhere to a standardized way of doing things — resulting in cross-compatibility becoming easier for content makers and there no longer being the need to build multiple versions of the same website.

**Note**: *There are still a number of nuances about cross-compatibility amongst browsers. Even today, over 20 years since standards were introduced, we aren’t quite at “one-size fits all” just yet.*

[Learn how event teams are using Hopin to delight their attendees, speakers, and sponsors.](https://cats.smashing.services/ball?uri=https://bit.ly/3l2gmqj)

[CREATE EVENT](https://cats.smashing.services/ball?uri=https://bit.ly/3l2gmqj)

Here’s a quick look at some of the key moments in the history of web browser development:

| **Year** | **Key moments** |
| --- | --- |
| 1990 | Sir Tim Berners Lee releases the WorldWideWeb, the first way in which to browse the web. |
| 1992 | MidasWWW was developed as another WWW browser, which included a source code viewer. |
| 1992 | Also in 1992 Lynx was released, which was a text-based web browser, it could not display images or any other graphic content. |
| 1993 | NCSA Mosaic was released, this is the browser that is credit for being the first to popularize web browsing as it allowed the display of image embedded within text. |
| 1995 | Microsoft released Internet Explorer, previously Cello or Mosaic browsers were used on Windows products. |
| 1996 | Opera was released publicly, it was previously a research project for a Norwegian telecoms company Telnor. |
| 2003 | Safari was released by Apple, previously Macintosh computers shipped with Netscape Navigator or Cyberdog. |
| 2004 | In the wake of Netscape Navigator’s demise, Firefox was launched as a free, open-source browser. |
| 2008 | Chrome was launched by Google and within six years grew to encompass the majority of the browser market. |
| 2015 | Microsoft released Edge, the new browser for Microsoft, replacing Internet Explorer from Windows 10 onwards. |

*Source: “*[*Web Browsers: A Brief History*](https://www.telegraph.co.uk/technology/microsoft/11577364/Web-browsers-a-brief-history.html)*” by Rhiannon Williams*

**Why We Need Standards**

Knowing a bit about the history of standards and why they were introduced, we can start to see the benefits of having standards for the World Wide Web. But why is it important that we continue to contribute to Web Standards? Here are just a few reasons:

**KEEPING THE WEB FREE AND ACCESSIBLE TO ALL**

Without the Web Standards community, browser makers would be the ones making decisions on what should and shouldn’t be features of the world wide web. This could lead to the web becoming a monopolized commodity, where only the largest players would have a say in what the future holds.

**HELPING MAKE SOURCE CODE SIMPLER; REDUCING DEVELOPMENT AND MAINTENANCE TIME**

As more browsers appeared and browser makers began to diversify in their approach, it became more and more difficult to create content that would be served in the same way across multiple browsers. This increased the amount of work required to make a fully compatible website, including bloating the source code for a web page. As developers today we still have to do the odd include [X script] so this works on [X web browser], but without Web Standards, this would be much worse.

**MAKING THE WEB A MORE ACCESSIBLE PLACE**

Web standards help to standardize the way in which a website can interact with assistive technologies. Meaning that browser makers and web developers can incorporate instructions into their pages which can be interpreted by assistive technologies to maintain a common (or sometimes better) end-user experience.

**ALLOWING FOR BACKWARD COMPATIBILITY AND VALIDATION**

Web standards have created a foundation which allows for new websites, that comply with standards, to work with older browser versions. This idea of backward compatibility is super important for keeping the web accessible. It doesn’t guarantee older browsers will show your content exactly as you expect, but it will ensure that the structure of the web document is understood and displayed accordingly.

**HELPING MAINTAIN BETTER SEO (SEARCH ENGINE OPTIMIZATION)**

Another of the major hidden benefits (at the time that Web Standards was first introduced) was that a Web Standards compliant website was more discover-able by search engines. This became more evident when the Google search became the major player in the search engine world in the early 2000s.

**CREATING A POOL OF COMMON KNOWLEDGE**

A world with web standards creates a place in which a set of rules exists, rules that every developer can follow, understand and become familiar with. In theory, this means that one developer could build a website that complies with standards and another developer could pick up where the former left off without much trouble. In reality, standards provide the foundation for this; but the idea relies heavily on developers writing well-documented code.

**Who Decides On What Becomes A Web Standard?**

Standards are created by people. In the web and Internet space, there is a strong culture of consensus — which means a lot of talking and a lot of discussions.

The groups through which standards are developed are sometimes referred to as “Standards Development Organisations” or **SDOs**. Key SDOs in the web space include the Internet Engineering Task Force (IETF), the World Wide Web Consortium (W3C), the WHATWG, and ECMA TC39. Historically there were also groups like the Web Standards Project (WaSP), that advocated for Web Standards to be adopted by organizations.

The groups that work on the Internet and Web Standards generally operate under a royalty-free regime. That means when you make use of a web standard you don’t have to pay anyone — like someone who might hold a relevant patent. Whilst the idea that you might have to pay royalties to someone to build a web browser or website might seem absurd right now, it wasn’t too long ago that [organizations like BT were trying to assert ownership of the concept of the hyperlink](https://www.zdnet.com/article/bt-loses-hyperlink-patent-case/). Standards organizations like the ones listed below help keep the web free (or free from licensing fees at least).

**WHAT IS IETF?**

The [IETF](https://ietf.org/) is the grandparent of Internet standards organizations. It’s where underlying Internet technologies like TCP/IP (Transmission Control Protocol/Internet Protocol) and DNS (Domain Name System) are standardized. Another key technology developed in IETF is something called Hyper-Text Transport Protocol (HTTP) which you may have heard of.

If you’ve been paying attention to the rise of [HTTP2](https://http2.github.io/) and the subsequent development of (UDP-based) [HTTP3](https://www.zdnet.com/article/http-over-quic-to-be-renamed-http3/), this is where that work happens. Most of the work in IETF is focused on the lower levels of the [Open Systems Interconnection model](https://en.wikipedia.org/wiki/OSI_model).

**WHAT IS W3C?**

The World Wide Web Consortium (W3C) is an international community where [member organizations](https://www.w3.org/Consortium/Member/List), a full-time [staff](https://www.w3.org/People/), invited experts and the public work together to develop Web Standards. Led by Web inventor and Director Tim Berners-Lee and CEO Jeffrey Jaffe, W3C’s mission is to lead the Web to its full potential.

The community was founded in 1994 at MIT (Massachusetts Institute of Technology) in collaboration with [CERN](https://home.cern/). At the time of this post, W3C has [475 member companies and organizations](https://www.w3.org/Consortium/Member/List) and exists as a consortium between 4 academic institutions: MIT (USA), ERCIM (France), KEIO University (Japan) and Beihang University (China).

Work in W3C happens in *working groups* and *community groups*. [Community groups](https://www.w3.org/community/) are where a lot of initial innovation happens around new web technologies. New web standards can be produced by community groups but they are officially seen as “pre-standard.” Community groups are open for anyone to participate, whether or not the organization you work for or are affiliated with is a W3C member.

W3C working groups are where new web standards are officially minted. [Working groups](https://www.w3.org/Consortium/activities) usually start with a submission of a standard, often something that is already shipping in some browsers. However, technical work on refining these standards happens within these groups before the standard goes for final approval as a “W3C Recommendation.” By the time something reaches “recommendation” phase in W3C, it’s most often implemented and in wide use across the web.

Working groups are more difficult for people who are not affiliated with a member organization to become a part of. However, you may [become an invited expert](https://www.w3.org/2004/08/invexp.html) to a group. One reason why working groups are a little more difficult to join and operate with more process is that they also act as an intellectual property holder  —  through joining a W3C working group organizations and companies agree to the royalty-free licensing laid out in [W3C’s patent policy](https://www.w3.org/Consortium/Patent-Policy/).

W3C Advisory Board member [Natasha Rooney](https://twitter.com/thisNatasha) has put together a great document, [W3C Process Document for Busy People](https://github.com/w3c/wg-effectiveness/blob/master/process.md), that explains a lot of the ins and outs of working in W3C.

**WHAT IS THE WHATWG?**

The [WHATWG](https://whatwg.org/) was originally a splinter group from the W3C. It was formed in 2007 because some browser vendors didn’t agree with the direction in which the W3C was pushing HTML. WHATWG continues to be the place where HTML is developed and evolved. However, the community of participation in the HTML specification still includes many people from the W3C community, and many WHATWG-affiliated people participate in W3C working groups.

At the time of this post, the relationship between the W3C and the WHATWG remains in flux. From a developer perspective, this doesn’t matter too much because developers can rely on resources like MDN to reflect the “truth” of which web technologies can be used in specific browsers. However, it has led to a lack of clarity, in terms of where to participate in the development of certain standards. WHATWG also has its own royalty-free license agreement  — [the WHATWG participation agreement](https://participate.whatwg.org/agreement).

**WHAT IS THE “WHY CG”?**

The [Web Incubator Community Group](http://wicg.io/) (WICG, pronounced *Why-CG*) is a special community group, within W3C, where some new and emerging web technologies are discussed and developed.

If you have a great idea for a new standard, a new feature for an existing standard or a new technology you think ought to be incorporated into the web, it’s worth checking here first to see if something like it is already being discussed. If it is, great! Jump into these discussions and lend your support. If not, then suggest it! That’s what this group is for.

**WHAT IS THE ECMA TC39?**

[Ecma](https://www.ecma-international.org/) is a standards organization for information and communication systems, which was founded in 1961 to standardize computer systems in Europe. Its name comes from being previously known as the “European Computer Manufacturers Association” but it is now referred to as “Ecma International  —  European association for standardizing information and communication systems” since the organization went global in 1994.

The ECMA-262 standard outlines the ECMAScript Language Specification, which is the standardized specification of the scripting language known as JavaScript. There are ten editions of ECMA-262 that have been published ([the tenth edition](http://www.ecma-international.org/ecma-262/9.0/index.html) was published in June 2018).

TC39 (Technical Committee 39) is the committee that evolves JavaScript. Like the other groups listed here, its members are companies which include most of the major browser makers. The committee has regular meetings which are attended by delegates sent from the member organizations and also by invited experts. The TC39 operates on achieving consensus, as with many of the other groups, and the agreements made often lead to obligations for its members (in terms of future features that member organizations will need to implement). [The TC39 process](http://2ality.com/2015/11/tc39-process.html) includes accelerating proposals through a set of stages, the progression of a proposal from one stage to the next must be approved by the committee.

**WHAT WAS THE WEB STANDARDS PROJECT?**

[The Web Standards Project](https://www.webstandards.org/) was formed in 1998 as a resistance to the feature face-off happening between browsers in the 90s; with a primary goal of getting browser makers to comply with the standards set forth by the W3C.

As the organization grew and the browser wars ended, the project began to shift focus. The group began working with browser makers on improving their standards support, consulting software makers that created tooling for website creation and educating web designers and developers on the importance of web standards. The last of these points, resulted in the creation of the [InterAct web curriculum framework](https://teach.webstandards.org/) which is now maintained by W3C.

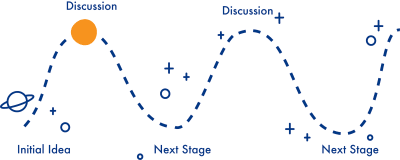
The Web Standards Project ceased to be active in 2013. A [final blog post](https://www.webstandards.org/2013/03/01/our-work-here-is-done/index.html) was created on March 1st that gives thanks to the hard work of the members and supporters of the project. In the closing remarks of this post, readers are reminded that the job of the Web Standards Project is not entirely over, and that the responsibility now lies with thousands of developers who continue to care about ensuring the web remains a free, open, inter-operable and accessible resource.

**How Does Something Become A Web Standard?**

So, how are standards made? The short answer is through LOTS of discussions.

Proposals for new standards usually start as a discussion within a community group (this is especially the case in W3C) or through issues raised on the relevant GitHub repository.

Across the different SDOs, there seems to then be a common theme of ascension; after the discussion has begun, it then moves up within the organization, and at each level, a deciding committee needs to reach a consensus to approve the elevation of that discussion. This is repeated until the discussion becomes a proposal, then that proposal becomes a draft and the draft goes on to become an official standard.

[](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/8f1a267f-8420-4475-915c-1650e27a4940/web-standards-web-standard-trajectory.png)([Large preview](https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/8f1a267f-8420-4475-915c-1650e27a4940/web-standards-web-standard-trajectory.png))

Now as previously mentioned, when something isn’t an official standard, this does not necessarily mean that it is not in use within some browsers. In fact, by the time something becomes a standard, it is likely to already have widespread use across many of the available browsers. In this instance, the role of the standard is part of the normalizing and adoption process for new features; it sets out the expected use for something and then outlines how browser makers and developers can conform to this expectation.

**WHAT IS TPAC?**

Every year, W3C holds one massive event, a week-long multi-group meeting punctuated by a one-day unconference on the Wednesday (the Technical Plenary) combined with a meeting of its Advisory Committee (a group consisting of one person for every organization or company that is a W3C member). Put Technical Plenary and Advisory Committee together, and you get TPAC (often pronounced *tee-pac*). Although it’s a W3C-run event, you will often find people “from” WHATWG, IETF or TC39 here as well.

This past year, Samsung Internet people came together to participate in TPAC. We also sponsored diversity scholarships which are intended to bring people from under-represented groups to TPAC and to the Web Standards community.