

WEB STANDARDS AND ACCESSIBILITY GUIDELINES

SCC.306 Guest Lecture

Lancaster University



B B C

REC-html32

HTML 3.2 Reference Specification

W3C Recommendation 14-Jan-1997
Superseded 15-March-2018

Author: [Dave Raggett](mailto:dsr@w3.org) (dsr@w3.org)

Status of this document

This specification is a W3C Recommendation. It is reviewed by the W3C HTML Working Group and other interested parties and has been endorsed by the W3C Executive Committee as a recommendation. This specification is a stable document and is intended to remain in place of this Working Draft. New implementations should follow the latest version of the HTML specification.

A current W3C Recommendation and other technical documents can be found at <https://www.w3.org/TR/>.

Abstract

The HyperText Markup Language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML version 3.2. HTML 3.2 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 2.0 (RFC 1866).

Contents

- [Introduction to HTML 3.2](#)
- [HTML as an SGML application](#)
- [The Structure of HTML documents](#)
- [The HEAD element and its children](#)
- [The BODY element and its children](#)
- [Sample SGML Open Catalog for HTML 3.2](#)
- [SGML Declaration for HTML 3.2](#)
- [HTML 3.2 Document Type Definition](#)
- [Character Entities for ISO Latin-1](#)
- [Table of printable Latin-1 Character codes](#)
- [Acknowledgements](#)
- [Further Reading ...](#)

Introduction to HTML 3.2

HTML 3.2 is W3C's specification for HTML, developed in early '96 together with vendors including IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass, and Sun Microsystems. HTML 3.2 adds widely deployed features such as tables, applets and text flow around images, while providing full backwards compatibility with the existing standard HTML 2.0.



Josh Tumath (he/him)

Senior Software Engineer Design System

B B C



WE WANT YOU

<https://www.bbc.co.uk/careers/>

B B C



Aims for this lecture

**That you will be
empowered to...**

...read a HTML or CSS
specification

...contribute to Web
standards yourself,
someday

Coming up...

1. What is the Web?
2. The standards organisations
3. Specifications you need to know about
4. Why do we need Web standards?
5. Accessibility guidelines

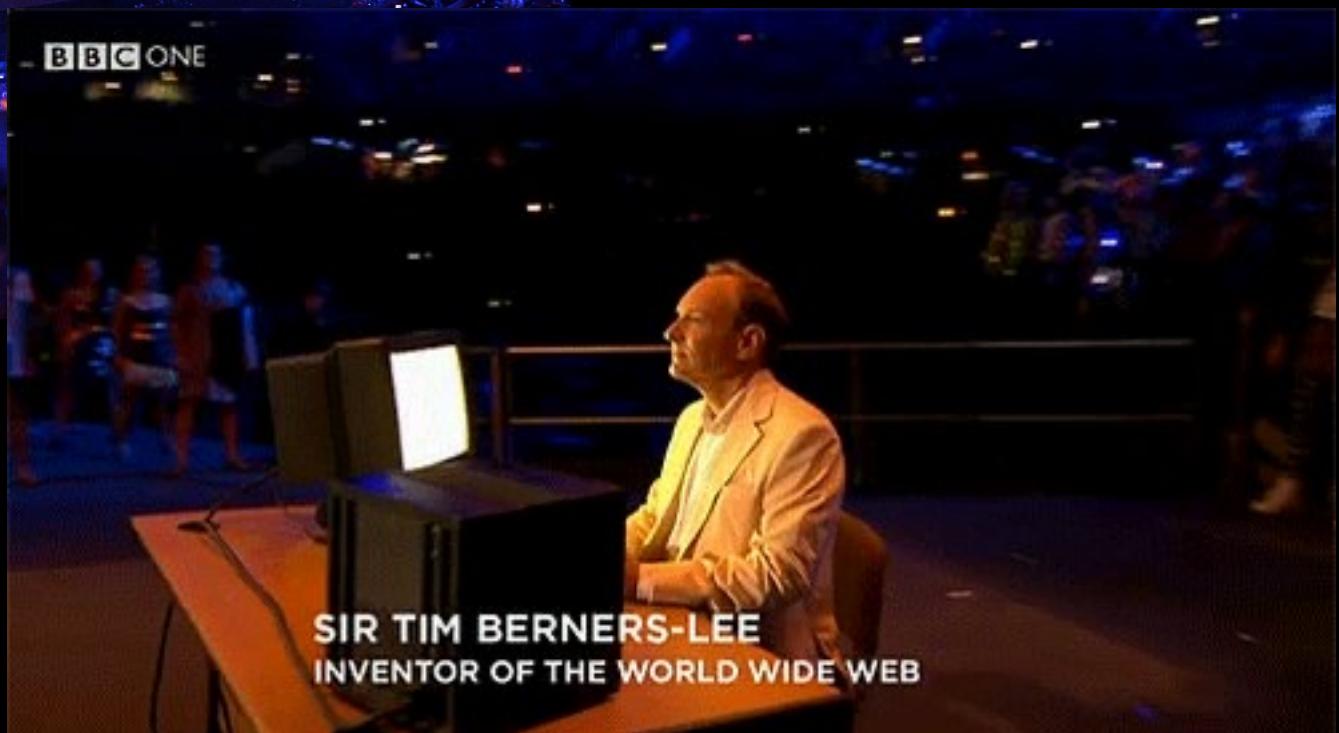
What is the Web?



Image: Inavate

B B C

@joshtumath



World Wide Web

A collection of
technologies

Web browser

HTML

URL

HTTP

Web server

World Wide Web

The first website



The screenshot shows a web browser window titled "The World Wide Web project". The URL in the address bar is "info.cern.ch/hypertext/WWW/TheProject.html". The main content area displays the text "World Wide Web" followed by a paragraph about the W3 project. Below this, there is a list of links under various categories such as "What's out there?", "Help", "Software Products", "Technical", "Bibliography", "People", "History", "How can I help?", and "Getting code".

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#) Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#) on the browser you are using

[Software Products](#) A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#) Details of protocols, formats, program internals etc

[Bibliography](#) Paper documentation on W3 and references.

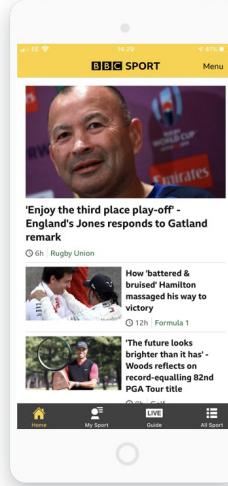
[People](#) A list of some people involved in the project.

[History](#) A summary of the history of the project.

[How can I help ?](#) If you would like to support the web..

[Getting code](#) Getting the code by [anonymous FTP](#) , etc.

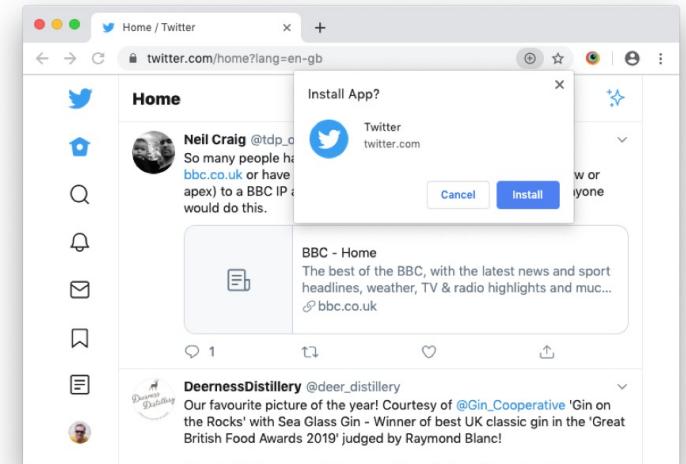
Now the Web is everywhere!



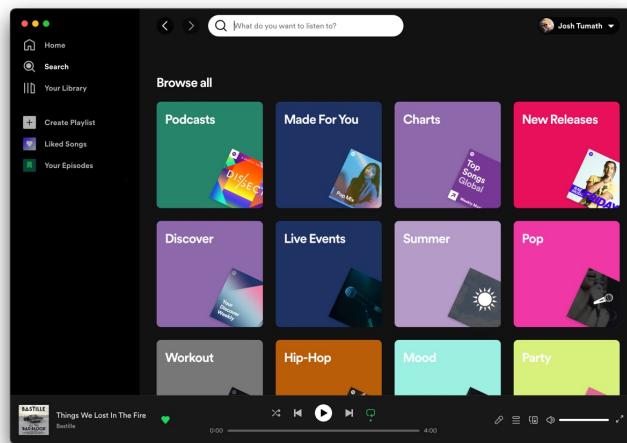
Smartphone apps



Smart TV apps



Progressive Web Apps



Desktop apps

Live TV graphics
'the lower thirds'

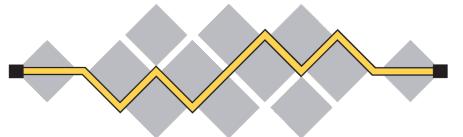


@joshtumath

The standards organisations

Standards organisations let people from industry come together to write *specifications* for *interoperable code*

The standards bodies



I E T F®

HTTP, QUIC, TLS...



HTML 4, CSS, SVG,
XML, WCAG, PNG...



@joshtumath



WHATWG

HTML Living Standard,
DOM, URL, Fetch
API, Fullscreen API...



EcmaScript
(i.e. JavaScript)

W3C's process

Many Working Groups



Working Group



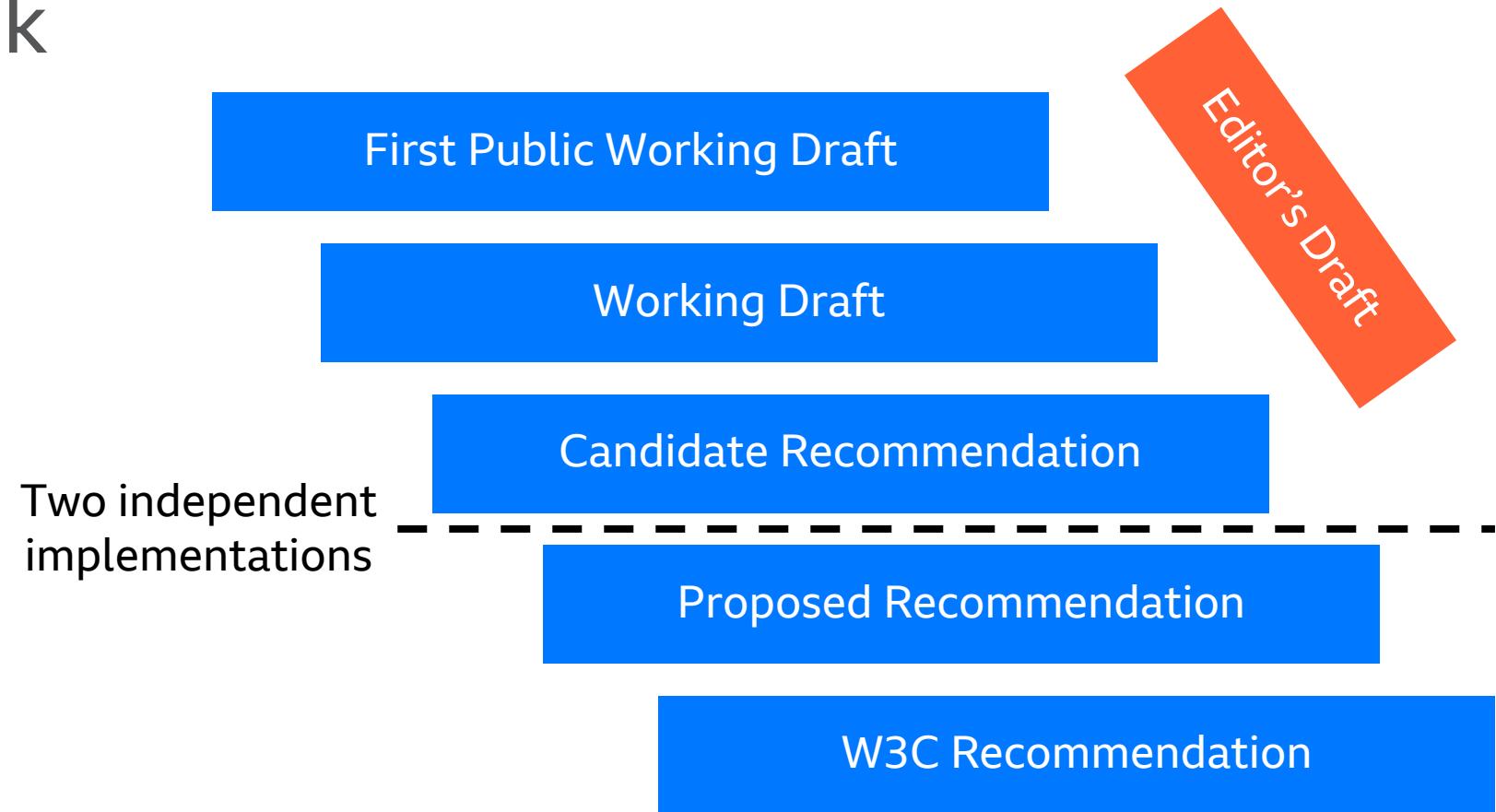
Charter

W3C's process

Standards track



Working Group



WHATWG's process

One Working Group, many Workstreams

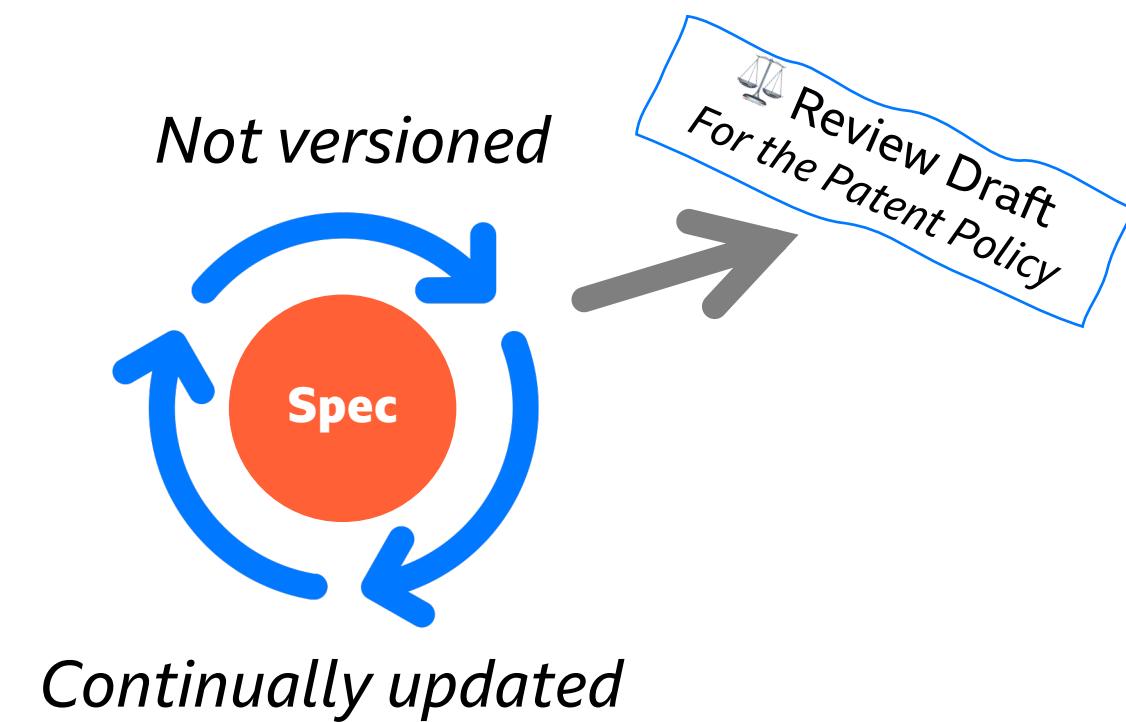


WHATWG's process

Living Standards



Workstream



Ecma International's process

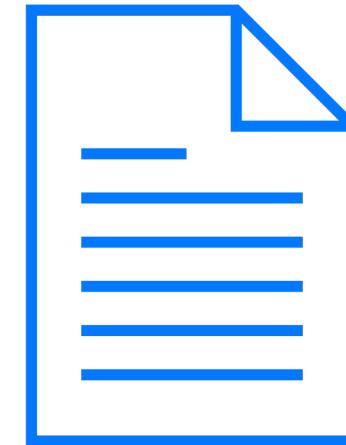
Committees



Committee



Ecma Members



Programme of work

TC39 committee's process

Standards track



TC39 committee

Stage 0 – Strawperson Proposal
Community shares idea for a feature

Stage 1 – Proposal
Formal proposal made to TC39

Stage 2 – Draft
Draft of addition to spec written

Stage 3 – Candidate
Almost ready; feedback from implementation

Stage 4 – Finished
Ready to be included in latest spec

Specifications you need to know about

HTML



@joshtumath

HTML

Timeline of specs

Nov
1995

HTML 2



Jan
1997

HTML 3.2



Dec
1997

HTML 4

- Strict
- Transitional
- Frames



Oct
2014

HTML 5



CHECK OUT THAT GAP!
What the heck happened here?

**So, let's go on a really boring
tangent and find out...**

XHTML



The ‘Web 2.0’
phenomenon



XHTML

Timeline of specs

Jan
2000



May
2001



Never

XHTML 1.0

- Strict
- Transitional
- Frames



XHTML 1.1



XHTML 2

- XForms
- XFrames
- XML Events



was formed



HTML Living Standard



@joshtumath

‘This specification should be read like all other specifications. First, it should be read cover-to-cover, multiple times. Then, it should be read backwards at least once. Then it should be read by picking random sections from the contents list and following all the cross-references.’

HTML specification



Demo: How to read the HTML Living Standard

<https://html.spec.whatwg.org/multipage/>

Areas we'll look at:

- 4. All the HTML elements
- 15. Rendering



@joshtumath

HTML

Living Standard — Last Updated 27 October 2022

Table of contents

- 1 Introduction
- 2 Common infrastructure
- 3 Semantics, structure, and APIs of HTML documents
- 4 The elements of HTML
- 5 Microdata
- 6 User interaction
- 7 Loading web pages
- 8 Web application APIs
- 9 Communication
- 10 Web workers
- 11 Worklets
- 12 Web storage
- 13 The HTML syntax
- 14 The XML syntax
- 15 Rendering
- 16 Obsolete features
- 17 IANA considerations
- Index
- References
- Acknowledgments
- Intellectual property rights

Full table of contents

- 1 Introduction

[File an issue about the selected text](#)

How to contribute

HTML Living Standard

<https://github.com/whatwg/html/blob/main/CONTRIBUTING.md>



@joshtumath

CSS



@joshtumath



Jen Simmons

My apparent mission to meet everyone in the CSSWG



Rachel Andrew



Adam Argyle



Una Kravets



Tab Atkins Jr

CSS

Timeline of specs

Dec
1996

CSS 1



May
1998

CSS 2



June
2011

CSS 2.1



Ongoing

Split into multiple
specs



It's not as bad as it looks

Demo: How to read a CSS specification

<https://www.w3.org/Style/CSS/current-work>

Areas we'll look at:

- The list of CSS specs
- CSS Scroll Snap Module Level 1
- Property definition boxes
- Example boxes

CSS Scroll Snap Module Level 1

W3C Candidate Recommendation Snapshot, 11 March 2021

This version:

<https://www.w3.org/TR/2021/CR-css-scroll-snap-1-20210311/>

Latest published version:

<https://www.w3.org/TR/css-scroll-snap-1/>

Editor's Draft:

<https://drafts.csswg.org/css-scroll-snap-1/>

Previous Versions:

<https://www.w3.org/TR/2019/CR-css-scroll-snap-1-20190319/>

<https://www.w3.org/TR/2019/CR-css-scroll-snap-1-20190131/>

<https://www.w3.org/TR/2018/CR-css-scroll-snap-1-20180814/>

<https://www.w3.org/TR/2017/CR-css-scroll-snap-1-20171214/>

<https://www.w3.org/TR/2017/CR-css-scroll-snap-1-20170824/>

<https://www.w3.org/TR/2017/CR-css-scroll-snap-1-20170209/>

<https://www.w3.org/TR/2016/CR-css-scroll-snap-1-20161020/>

<https://www.w3.org/TR/2016/WD-css-scroll-snap-1-20160623/>

<https://www.w3.org/TR/2016/WD-css-snappoints-1-20160329/>

<https://www.w3.org/TR/2015/WD-css-snappoints-1-20150326/>

Implementation Report:

<https://wpt.fyi/results/css/css-scroll-snap>

Test Suite:

http://test.csswg.org/suites/css-scroll-snap-1_dev/nightly-unstable/

Issue Tracking:

[CSSWG Issues Repository](#)

Editors:

Matt Rakow (Microsoft)

Jacob Rossi (Microsoft)

[Tab Atkins-Bittner](#) (Google)

[Elika J. Etemad / fantasai](#) (Invited Expert)

Suggest an Edit for this Spec:

[GitHub Editor](#)

How to contribute

CSS specs

<https://github.com/w3c/csswg-drafts/blob/main/CONTRIBUTING.md>

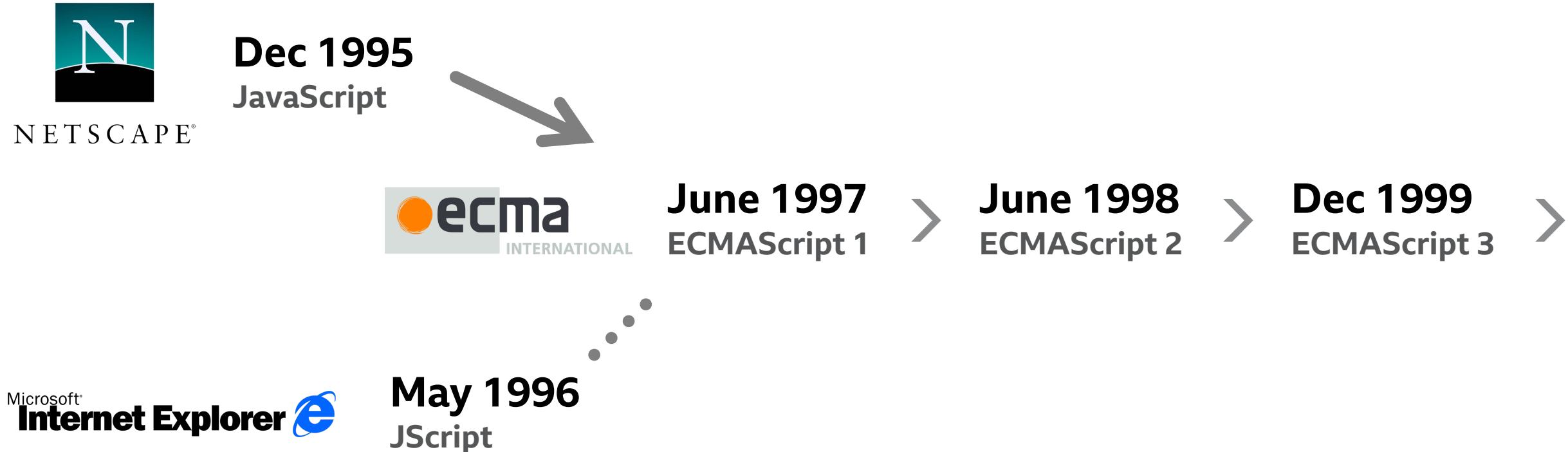


@joshtumath

EcmaScript

EcmaScript

Timeline of specs



EcmaScript

Timeline of specs

CHECK OUT THAT GAP!
What the heck happened here?

June 1998
ECMAScript 2

Dec 1999
ECMAScript 3

Dec 2009
ECMAScript 5

June 2011
ECMAScript 5.1

June 2015
ECMAScript 2015

*Yearly release
cycle, ever since*



@joshtumath

~~Demo~~: How to read the EcmaScript 2022 Language Specification

Just don't. (Use MDN instead.)

It's only written for JS engine developers

- = et `errors` be `F.[[Errors]]`.
 - = et `promiseCapability` be `F.[[Capability]]`.
7. Let `remainingElementsCount` be `F.[[RemainingElements]]`.
 8. Set `errors[index]` to `x`.
 9. Set `remainingElementsCount.[[Value]]` to `remainingElementsCount.[[Value]] - 1`.
 10. If `remainingElementsCount.[[Value]]` is 0, then
 - a. Let `error` be a newly created `AggregateError` object.
 - b. Perform `! DefinePropertyOrThrow(error, "errors", PropertyDescriptor { [[Configurable]]: true, [[Enumerable]]: false, [[Writable]]: true, [[Value]]: CreateArrayFromList(errors) })`.
 - c. Return `? Call(promiseCapability.[[Reject]], undefined, « error »)`.
 11. Return `undefined`.

The "length" property of a `Promise.any` reject element function is `1_F`.

27.2.4.4 `Promise.prototype`

The initial value of `Promise.prototype` is the `Promise prototype object`.

This property has the attributes `{ [[Writable]]: false, [[Enumerable]]: false, [[Configurable]]: false }`.

27.2.4.5 `Promise.race (iterable)`

The `race` function returns a new promise which is settled in the same way as the first passed promise to settle. It resolves all elements of the passed `iterable` to promises as it runs this algorithm.

1. Let `C` be the `this` value.
2. Let `promiseCapability` be `? NewPromiseCapability(C)`.
3. Let `promiseResolve` be `Completion(GetPromiseResolve(C))`.
4. `IfAbruptRejectPromise(promiseResolve, promiseCapability)`.
5. Let `iteratorRecord` be `Completion(GetIterator(iterable))`.
6. `IfAbruptRejectPromise(iteratorRecord, promiseCapability)`.
7. Let `result` be `Completion(PerformPromiseRace(iteratorRecord, C, promiseCapability, promiseResolve))`.
8. If `result` is an abrupt completion, then



@joshtumath

How to contribute

EcmaScript.next

<https://github.com/tc39/ecma262/blob/main/CONTRIBUTING.md>



@joshtumath

Why do we need standards?



B B C

@joshtumath

A history lesson...



NETSCAPE®

VS

Microsoft®
Internet Explorer 

- **Bugs**
- **Proprietary features**
- **Poor interoperability**

IE eventually dominates

Microsoft wins the browser wars

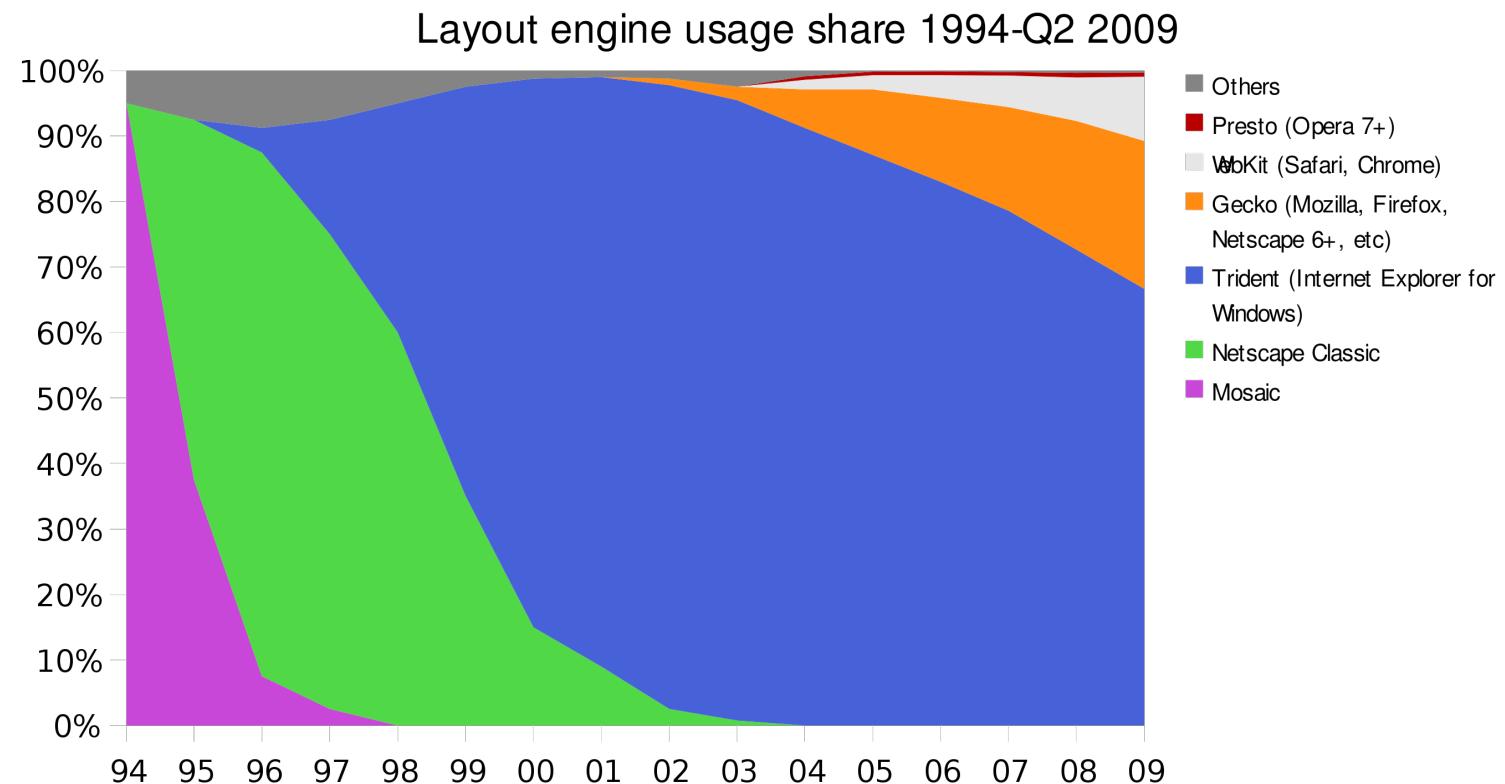


Image: [Public Domain](#)

Windows Vista is delayed

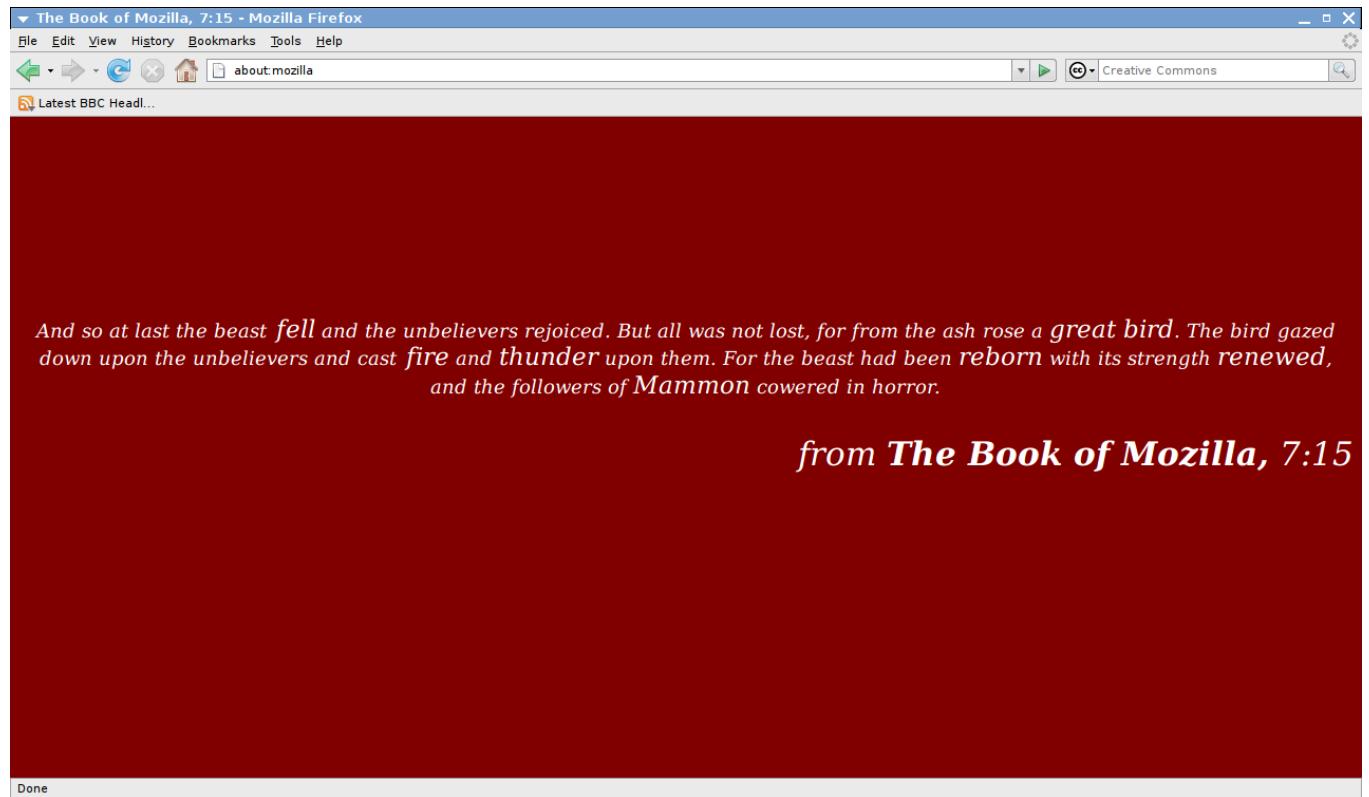
So we're stuck with IE6 for 5 years





B B C

@joshtumath



Firefox takes back market share

And Microsoft fails to innovate

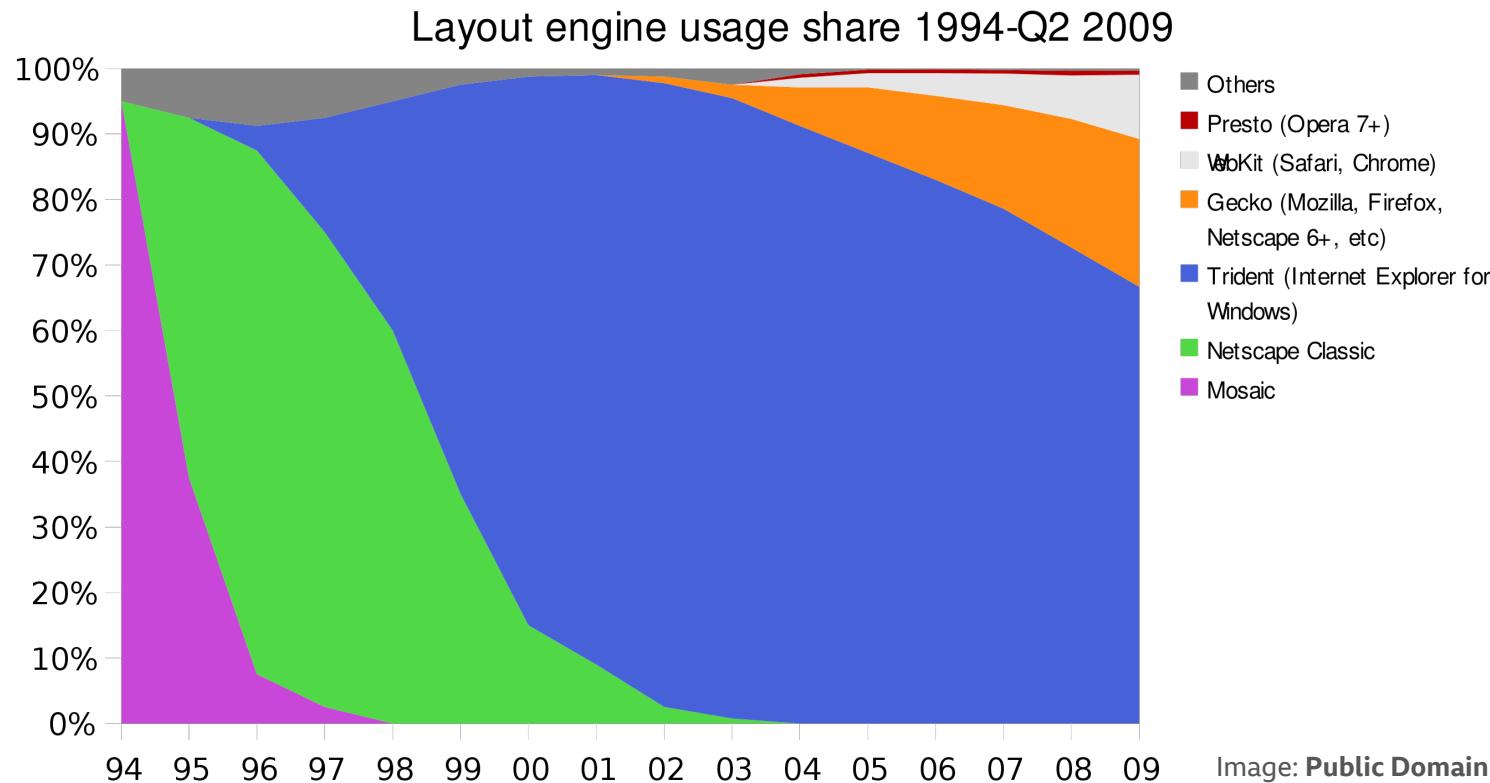


Image: Source unknown



B B C

@joshtumath

But that would never
happen today, right?

Safari

Abusing its monopoly on iOS

- Used to have many proprietary features and CSS properties
- Avoids adding new features for Progressive Web Apps
- Apple doesn't allow competing Web browser engines on the iOS app store (they all have to use WebKit)

HEY GANG, LET'S SEE
WHO'S REALLY BEHIND
THIS WEB BROWSER

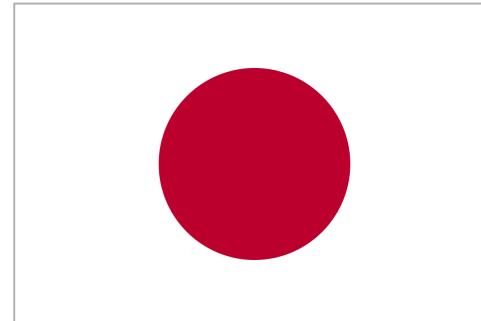


Safari

Governments getting involved



open-web-advocacy.org



Without standards and healthy competition, a company can gain control over the platform

Accessibility guidelines



Web Accessibility
Initiative WAI

WCAG 2.1

WCAG 2.1

Conformance levels

Level A

The bare minimum required

Web page satisfies all Level A Success Criteria

Level AA

Recommended

Web page satisfies all Level A and AA Success Criteria

Level AAA

Not always possible to achieve

Web page satisfies all Level A , AA and AAA Success Criteria

Demo: How to read WCAG 2.1

<https://www.w3.org/TR/WCAG21/>

Areas we'll look at:

- The various guidelines
- The quick reference
- Guideline 1.4.3: Contrast (Minimum) – **Level AA**



@joshtumath

Web Content Accessibility Guidelines (WCAG) 2.1

W3C Recommendation 05 June 2018

This version:

<https://www.w3.org/TR/2018/REC-WCAG21-20180605/>

Latest published version:

<https://www.w3.org/TR/WCAG21/>

Latest editor's draft:

<https://w3c.github.io/wcag/21/guidelines/>

Implementation report:

<https://www.w3.org/WAI/WCAG21/implementation-report/>

Previous version:

<https://www.w3.org/TR/2018/PR-WCAG21-20180424/>

Previous Recommendation:

<https://www.w3.org/TR/2008/REC-WCAG20-20081211/>

Editors:

[Andrew Kirkpatrick](#) (Adobe)

[Joshue O Connor](#) (Invited Expert, InterAccess)

[Alastair Campbell](#) (Nomensa)

[Michael Cooper](#) (W3C)

WCAG 2.0 Editors (until December 2008):

Ben Caldwell (Trace R&D Center, University of Wisconsin-Madison)

Loretta Guarino Reid (Google, Inc.)

Gregg Vanderheiden (Trace R&D Center, University of Wisconsin-Madison)

Wendy Chisholm (W3C)

John Slatin (Accessibility Institute, University of Texas at Austin)

Jason White (University of Melbourne)

Please check the [errata](#) for any errors or issues reported since publication.

See also [translations](#).

This document is also available in non-normative formats, available from [Alternate Versions of Web Content Accessibility Guidelines 2.1](#)

WCAG 3

Demo: APCA Contrast Calculator

<https://www.myndex.com/APCA/>

APCA Contrast Calculator

TEXT / ICON COLOR

#1234b0 • rgb(18,52,176)

1234b0

SAMPLE & Click To Swap

Contrast Lc 75.6

BACKGROUND COLOR

#e9e4d0 • rgb(233,228,208)

e9e4d0

All font sizes are in CSS px • Reference font shown with colors at actual size & weight

USAGE	200	300 (Light)	400 (Normal)	500	600	700 (Bold)
BODY TEXT OKAY	42px	24px	18px	16px	15px	14px
	Samp	The lazy grey dog slept in afternoon sun, but the fr	frolicked freely in the field flowers without a care in	The fox saw t	The fox saw th	The fox saw th
		afternoon sun, but the fr	frolicked freely in the field	flowers without a care in	flowers without a care in	flowers without a care in
		afternoon sun, but the fr	frolicked freely in the field	flowers without a care in	flowers without a care in	flowers without a care in
		afternoon sun, but the fr	frolicked freely in the field	flowers without a care in	flowers without a care in	flowers without a care in

Create a shareable URL of these colors

Generate a URL of This Color Pair

Accessible Contrast Relative to Font Size and Weight

APCA reports lightness contrast as an Lc value from **Lc 0** to **Lc 106** for dark text on a light background, and Lc 0 to Lc -108 for light text on a dark background (dark mode). The minus sign merely indicates negative contrast, which means light text on a dark background.

For resources, articles, and documentation, please [visit our page at GitHub](#).



@joshtumath

**You can't rely on WCAG or
automated tools alone to check if
you have good accessibility**

**Don't tack accessibility on at the
end of your project. It needs to be
considered from the very start.**

Conclusion



Tim Burners-Lee's vision for the Web is still relevant



Web Standards organisations have different maturation processes



Knowing how to read a spec is useful



Everyone can participate in a Web Standard – even you!



To protect our free and open Web, we need to understand its history



WCAG guidelines help us make sure our websites don't disable people



This lecture is too long