Web features, Baseline status, and standardization signals

W3C Breakouts Day 2024
12 March 2024

François Daoust - Patrick Brosset - Kadir Topal

Welcome to the breakout! 👋

- 🗣️ Presentation is a means to an end. Discussion is key!
- 🔇 Please mute when you're not speaking.
- ✋ Please use “Raise hand” feature in Zoom to raise a question or comment.
- ⏹️ Discussion will **not** be recorded.
- 📝 Discussion notes taken on IRC, `irc.w3.org`, channel `#web-features`.
- 🫡 Breakout operated under W3C’s Code of Ethics and Professional Conduct.

Web developers 🥷
& web features ⚙️

Web developers

Dos (*)

- Talk about **features** such as “Fetch priority”.
- Read **dedicated documentation** about features, e.g., on MDN.
- Look for **useful compatibility data** on MDN, Can I Use, etc.

Don’ts (*)

- Don’t care about how features get grouped into **specs**.
- Don’t read **specs**, which is OK, because they’re hard to read.
- Don’t look at the **standardization status** of specs to use a feature.

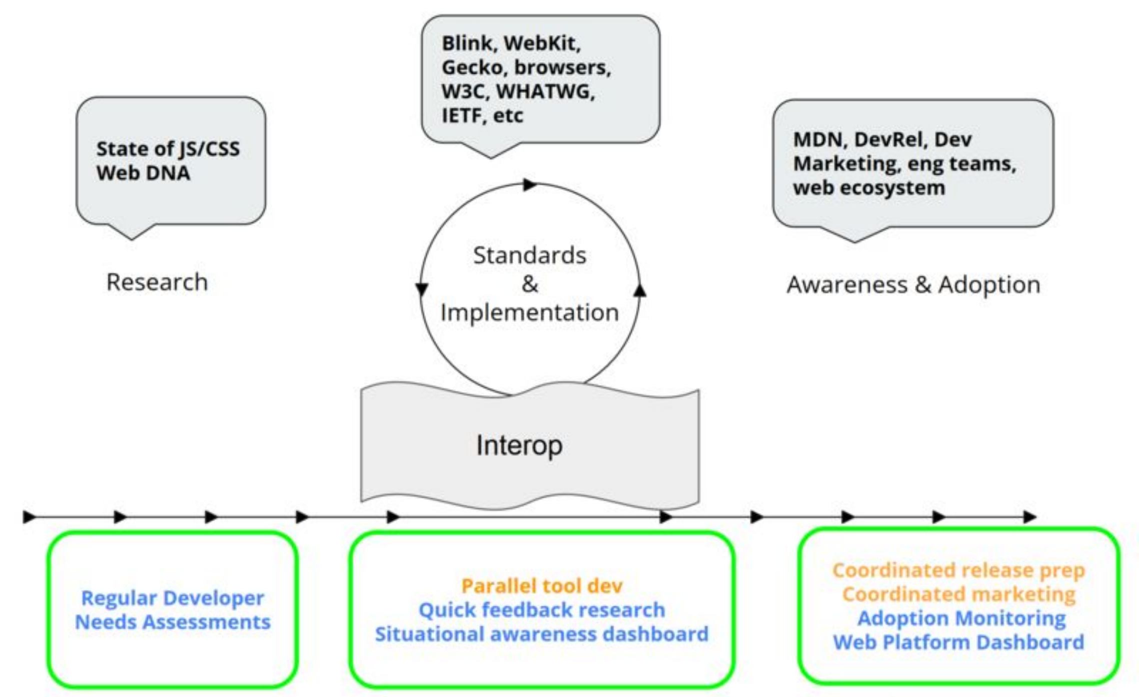
(*) Your mileage may vary…

Ideally, web features…

- **Identify** what web developers get and what they want from the web platform, in terms that speak to their needs and interests.
- **Track** some functionality **across sources** (surveys, docs, support data, specs, tests) and **over time**.
- **Capture interoperability information** that developers can rely on.

Examples: Aborting ongoing activities, Grids, Popover, WebTransport, etc.

Web feature phases



State of JS/CSS
Web DNA

Research

Blink, WebKit, Gecko, browsers, W3C, WHATWG, IETF, etc

Standards & Implementation

Interop

MDN, DevRel, Dev Marketing, eng teams, web ecosystem

Awareness & Adoption

Regular Developer Needs Assessments

Parallel tool dev
Quick feedback research
Situational awareness dashboard

Coordinated release prep
Coordinated marketing
Adoption Monitoring
Web Platform Dashboard

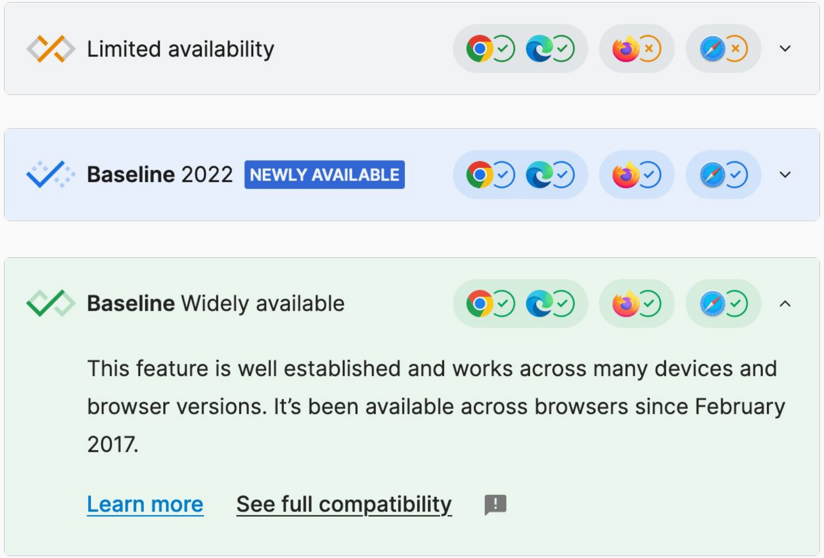
The web-features project

A **catalog** of web features developed by the WebDX Community Group to

- **Identify** capabilities of the web.
- **Track** features across phases.
- **Capture** *Baseline* availability across key browsers and releases through

Published as an NPM package.
See web-platform-dx/web-features repository.

Baseline status



Limited availability

Baseline 2022 NEWLY AVAILABLE

Baseline Widely available

This feature is well established and works across many devices and browser versions. It’s been available across browsers since February 2017.

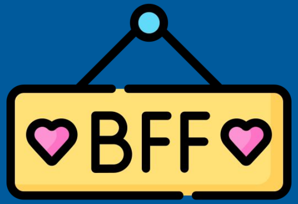
Learn more See full compatibility

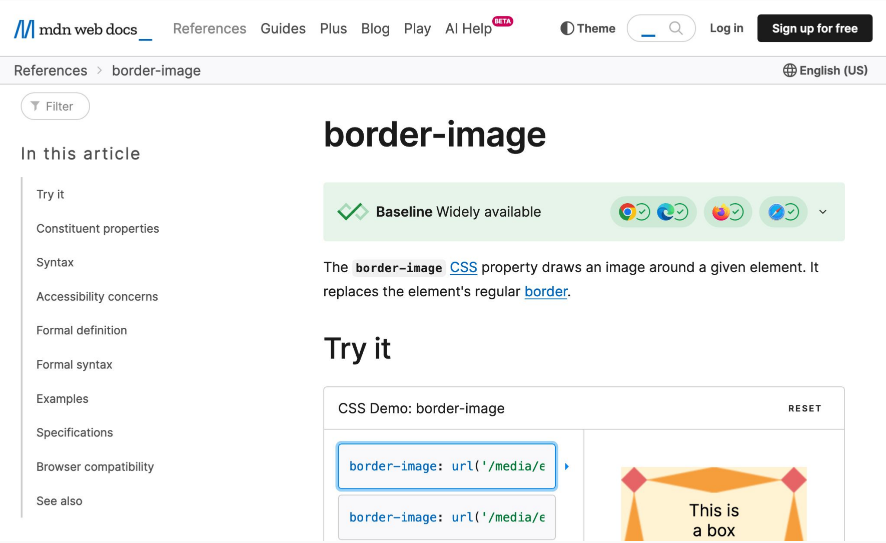
From “Baseline's evolution on MDN” - 5 december 2023
https://developer.mozilla.org/en-US/blog/baseline-evolution-on-mdn/

- Identify well supported features.
- Computed from support data in the MDN `browser-compat-data` (**BCD**) project.
- Two Baseline statuses:
  - **Baseline low**: available across latest versions of browsers in the core browser set (*)
  - **Baseline high**: widely available. Across browsers in the core browser set (*) for at least 30 months.

(*) Apple Safari (macOS, iOS), Google Chrome (desktop, Android), Microsoft Edge (desktop), Mozilla Firefox (Android, desktop)

Baseline in MDN and Can I Use





mdn web docs
References Guides Plus Blog Play AI Help BETA
Theme
Log in
Sign up for free
References > border-image
English (US)
Filter

In this article

- Try it
- Constituent properties
- Syntax
- Accessibility concerns
- Formal definition
- Formal syntax
- Examples
- Specifications
- Browser compatibility
- See also

border-image

✓ **Baseline** Widely available

The `border-image` CSS property draws an image around a given element. It replaces the element's regular border.

Try it

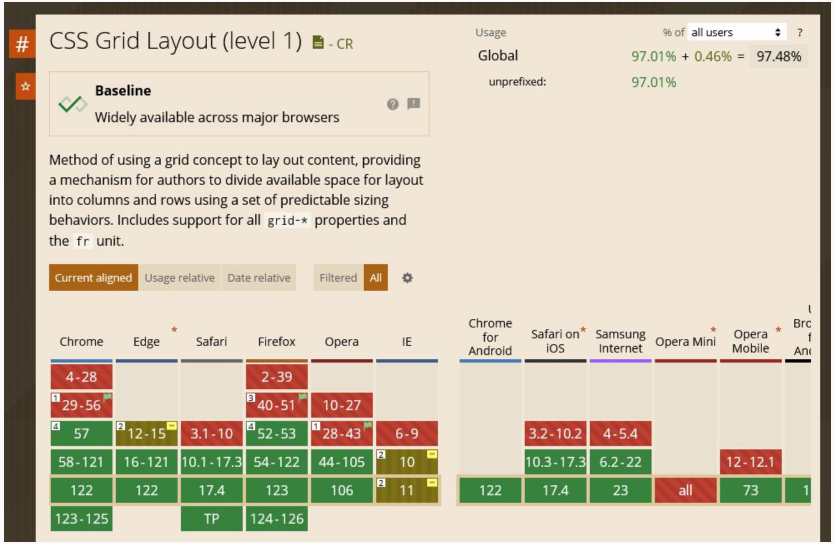
CSS Demo: border-image

RESET

```
border-image: url('/media/e
```

```
border-image: url('/media/e
```

This is a box



CSS Grid Layout (level 1) - CR

Baseline
Widely available across major browsers

Usage | % of all users | ?

Global	97.01% + 0.46% = 97.48%
unprefixed:	97.01%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	Bro f An
4-28			2-39								
[1] 29-56			[3] 40-51	10-27							
[4] 57	[2] 12-15	3.1-10	[4] 52-53	[1] 28-43	6-9		3.2-10.2	4-5.4			
58-121	16-121	10.1-17.3	54-122	44-105	[2] 10		10.3-17.3	6.2-22		12-12.1	
122	122	17.4	123	106	[2] 11	122	17.4	23	all	73	1
123-125		TP	124-126								

Other integrations

The WebDX CG is currently working with other projects to propagate identifiers provided by the web-features project, and improve tracking of features across sources:

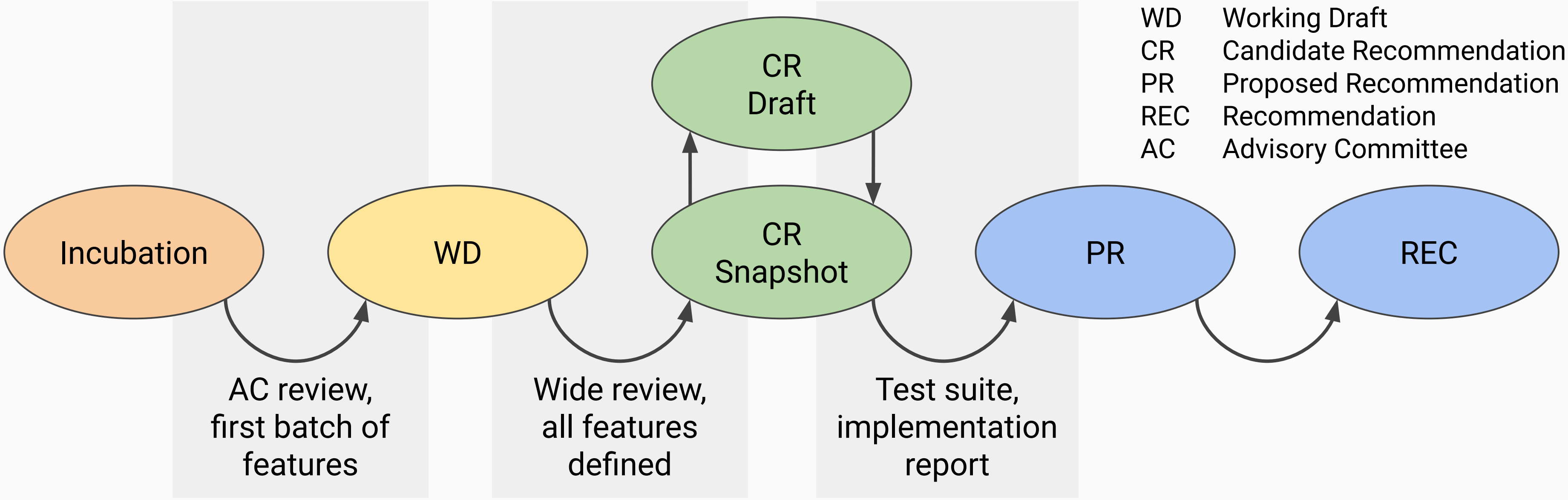
- Web Platform Tests
- State of * surveys
- …
- *What about integration in the standardization process?*

Web features
& Standardization

The W3C Process ❤️ features

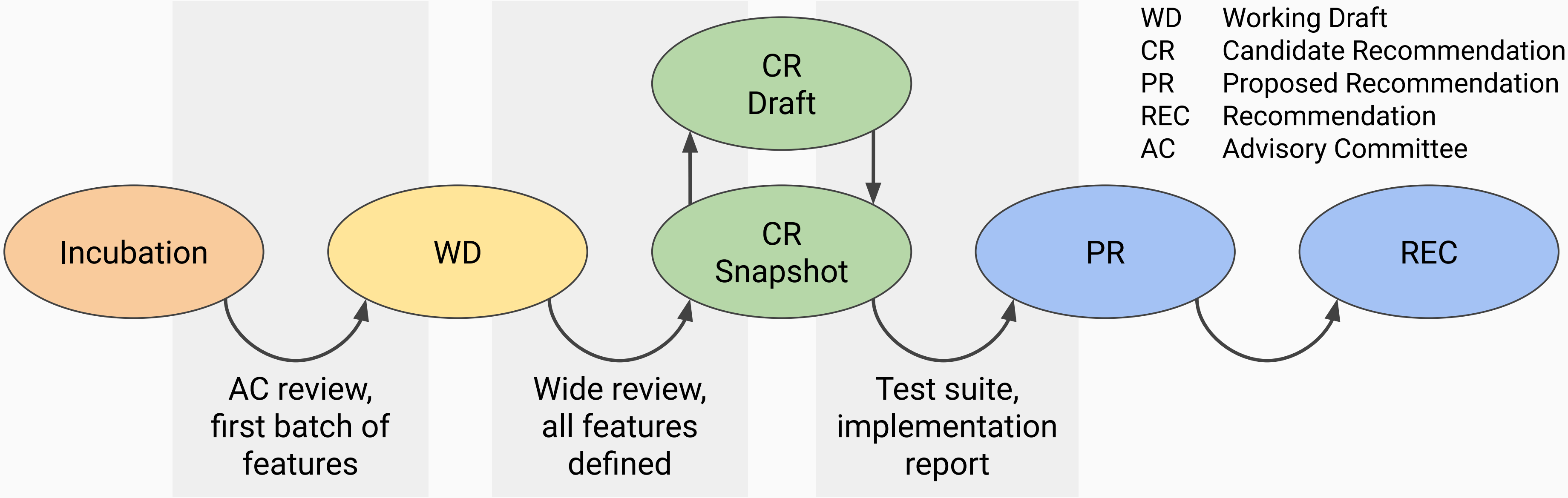
- It defines “**New features**” as “Changes that add new functionality, such as new elements, new APIs, new rules, etc.” (a.k.a. *class 4 changes*)
  https://www.w3.org/2023/Process-20231103/#class-4
- It uses features to assess “**adequate implementation experience**”
  https://www.w3.org/2023/Process-20231103/#implementation-experience
- It sets requirements for “**new features**” at certain maturity levels, e.g.,
  https://www.w3.org/2023/Process-20231103/#allow-new-features
- It allows “**features at risk**” in Candidate Recommendations
  https://www.w3.org/2023/Process-20231103/#at-risk

The Recommendation track



WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation
AC	Advisory Committee

The Recommendation track &
the Baseline status of spec features



Incubation

WD

CR
Draft

CR
Snapshot

PR

REC

AC review,
first batch of
features

Wide review,
all features
defined

Test suite,
implementation
report

WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation
AC	Advisory Committee

Not Baseline

Baseline low

Baseline high

Standardization signals?

Baseline may be used to **detect divergences** between the developer perspective and the standardization status of the underlying specs:

- **Late incubations**:
  Specs that define Baseline features but that are not on the Recommendation track.
- **Late Working Drafts**:
  Specs that define Baseline features but that are still Working Drafts.
- **Not-so-well supported Recommendations**:
  Recommendations that still define non-Baseline features.
- **Process assessment**:
  Interval between publication as Recommendation and features reaching Baseline status (note that interval may go both ways!).
  Evolution of the number of specs in the previous lists.

First stab at exploring standardization signals

Exploring standardization signals

- See code in `tidoust/web-features-standardization` repository and first report in issue #1.
- Combines Baseline info in `web-features` and spec status info in `browser-specs`, using BCD to map features to specs.
- Reports specs that may be worth looking into:
  - Late incubations
  - Late Working Drafts
  - Not-so-well supported Recommendations
- No analysis of interval between Rec and Baseline for now!
(info not directly available in `browser-specs`)

Highlights from the first report

Late incubations?

- Compression Streams
- CSS Environment Variables Module Level 1
- File and Directory Entries API
- Media Playback Quality
- `requestVideoFrameCallback()`
- The “threads” fork of WebAssembly JS Interface

Late Working Drafts?

- CSS Box Alignment Module Level 3
- CSSOM View Module (`scrollIntoView()` and `layerName`)
- Motion Path Module Level 1
- Selectors Level 4
- Server Timing
- Web Animations
- WebTransport (first batch of features)

Not-so-well supported Recommendations?

- Web Share API
- Web Cryptography API (some bits under `SubtleCrypto`)

Some known caveats

- **Features coverage** in `web-features` is far from complete.
  To extend coverage, code runs a second analysis considering each spec is a feature and computing Baseline info from BCD.
- **Signals are just signals**. They need to be analyzed.
  For instance, specs identified as potentially late may well define other less advanced features.
- **BCD keys often map to an unversioned spec URL**. The code uses the "current spec" in the series when that happens. That may not be correct.
- **Some Recommendations are missing** from `browser-specs` as the project focuses on the latest version in a series.

Some known reasons for delays

Out of incubation:

- Chartering a Working Group is hard.
- Hard to secure resources to move forward (chairs, editors, testers, etc.)
- Publication as First Public Working Draft triggers call for exclusions. Groups may delay publication until the spec contains enough features.

Out of Working Draft:

- Wide review is hard, especially when the spec keeps on moving.
- Groups use a more real-time feedback loop to inform and revise the spec based on implementation experience, and tend to delay publication of specs as Candidate Recommendation until the work is essentially done (scope, tests, implementations).

Discussion

Quick Q&A

- **What about computing interoperability from Web Platform Tests?**
  - Too fine-grained, always a test that fails somewhere.
  - Harder to map to features.
- **Is web-features really needed here?**
  - It conveys a developers perspective, which completes the standardization perspective with more practical “can use / can’t use” considerations.
  - It provides a mapping to BCD keys, allowing to exclude keys that are potentially less interoperable but also less interesting for main usage scenarios (e.g., event constructors).
  - It allows to distinguish features within a spec. A spec-based approach only sees the Web Speech API as a whole. Features see “Speech synthesis” and “Speech recognition” with very different compatibility statuses.

Possible discussion topics

- What additional data or tooling would make web-features a powerful tool for standardization?
- How would you like to see web-features data presented in standards work?
- On the meaning of “interoperability”
  - Different levels at which it can be evaluated
  - Example of accessibility support in assistive technology.

Thank you!

Attributions

Icons from www.flaticon.com:

- Brickwall, Question mark, Must have, Medal, Website, BFF, Hard work, Star, Timeline, Under construction icons created by Freepik
- Promise icons, Brainstorming icons created by Eucalyp
- Wish list icons created by monkik