

Introduction to the CSS Working Group

Rachel Andrew, CSS Print Workshop Prague 2020

What is the W3C?

The World Wide Web Consortium (W3C)

25 years old on 1st October 2019

“...create freely available and open standards that ensure that the Web remains open, accessible, and interoperable for everyone around the globe”

<https://www.w3.org/blog/2019/10/happy-25th-anniversary-world-wide-web-consortium/>

Members

Organizations become Members of the W3C

Working Groups

For example the **CSS Working Group**.

Accessibility Guidelines, Accessible Platform Architectures, Accessible Rich Internet Applications, Audio, Automotive, Browser Testing and Tools, CSS, Dataset Exchange, Decentralized Identifier, Devices and Sensors, Distributed Tracing, Education and Outreach, HTML, Immersive Web, Internationalization, JSON-LD, Media, Pointer Events, Publishing, Second Screen, Service Workers, SVG, Timed Text, Verifiable Claims, Web Application Security, Web Applications, Web Authentication, Web Fonts, Web of Things, Web Payments, Web Performance, Web Platform, Web Real-Time Communications, WebAssembly

Working Group members

People who are employed by a member organization can be joined to any working group.

Invited Experts

Not part of a member organization.

The CSS Working Group

This page is about the CSS Working Group.

For general information about CSS, see the [CSS Home Page](#).

For the group's meeting schedule, see the [internal page](#) [member-only link].

PARTICIPATING

The CSS working group discusses all technical matters on a public mailing list, www-style@w3.org. See “[If you want to help](#)” for more about that list.

The current draft CSS specifications are listed and explained on the “[current work](#)” page. In March 2008, the group started using [Tracker](#) to keep lists of [issues](#), [actions](#) and [group members](#). See also the [working group's charter](#) (version 2016–2019).

The group also maintains a [blog](#)...

... and if you want to know what it is like to be a member, read [Fantasai's](#) “[about:csswg](#).”

Name

Affiliation

Said Abou-Hallawa	Apple, Inc.
Rakina Amni	Google, Inc.
Rachel Andrew	Fronteers
Adam Argyle	Google, Inc.
Rossen Atanassov	Microsoft Corporation
Tab Atkins Jr.	Google, Inc.
Takao Baba	BPS Co., Ltd.
Mihail-Alexandru Balan	Adobe
Ashwin Balasubramaniyan	Magic Leap, Inc.
David Baron	Mozilla Foundation
Rafal Bartoszek	Viacom inc.
Daniel Bates	Apple, Inc.
Amelia Bellamy-Royds	W3C Invited Experts
Garrett Berg	Airbnb Inc
Christian Biesinger	Google, Inc.
Brian Birtles	W3C Invited Experts
Bert Bos	W3C Staff
Mike Bremford	BFO
Oriol Brufau	Igalia
Rick Byers	Google, Inc.
Bo Campbell	IBM Corporation
Tantek Çelik	Mozilla Foundation
cathie chen	Igalia
Junichi Chiba	BPS Co., Ltd.
Emilio Cobos Álvarez	Mozilla Foundation



Specification Process

Working Draft
Candidate Recommendation
Proposed Recommendation
W3C Recommendation

Why do things take **so long**?

Incomplete List of Mistakes in the Design of CSS

That should be corrected if anyone invents a time machine. :P

- `white-space: nowrap` should be `white-space: no-wrap`
 - and line wrapping behavior should not have been added to `white-space`
- `vertical-align` should not apply to table cells. Instead the CSS3 alignment properties should exist in Level 1.
- `vertical-align: middle` should be `text-middle` or `x-middle` because it's not really in the middle, and such a name would better describes what it does.
- Percentage heights should be calculated against `fill-available` rather than being undefined in auto situations.
- Table layout should be sane.
- Box-sizing should be `border-box` by default.
- `background-size` with one value should duplicate its value, not default the second one to `auto`. Ditto `translate()`.
- `background-position` and `border-spacing` (all 2-axis properties) should take `*vertical*` first, to match with the 4-direction properties like `margin`.
- The 4-value shorthands like `margin` should go counter-clockwise (so that the inline-start value is before the block-start value).
- `z-index` should be called `z-order` or `depth` and should Just Work on all elements (like it does on flex items).
- `word-wrap/overflow-wrap` should not exist. Instead, `overflow-wrap` should be a keyword on `'white-space'`, like `nowrap` (`no-wrap`).
- The top and bottom margins of a single box should never have been allowed to collapse together automatically as this is the **root of all margin-collapsing evil**.
- Partial collapsing of margins instead of weird rules to handle min/max-heights?
- Tables (like other non-blocks, e.g. flex containers) should form pseudo-stacking contexts.
- The `currentcolor` keyword should have a dash, `current-color`. Likewise all other color multi-word keyword names.
- There should have been a predictable color naming system instead of arbitrary X11 names.
- `border-radius` should have been `corner-radius`.
- Absolutely-positioned replaced elements should stretch when opposite offset properties (e.g. `left+right`) are set, instead of being start-aligned.
- The `hyphens` property should be called `hyphenate`. (It's called `hyphens` because the XSL:FO people objected to `hyphenate`.)
- `rgba()` and `hsla()` should not exist, `rgb()` and `hsl()` should have gotten an optional fourth parameter instead (and the alpha value should have used the same format as R, G, and B or S and L).

Lack of funding for work on
print specifications

Web browsers and Paged Media support

It's not just printing from the browser

Web developers are using headless Chrome to generate PDF documents and running into the same issues.

**ABOUT THE AUTHOR**


Rachel Andrew is not only Editor in Chief of Smashing Magazine, but also a web developer, writer and speaker. She is the author of a number of books, including ... [More about Rachel Andrew ...](#)

JUNE 19, 2019 • [15 comments](#)

How To Create A PDF From Your Web Application

[Edit in CMS](#)

QUICK SUMMARY ↗ *There is a wide variety of choices when it comes to creating a PDF from a web application. In this article, Rachel Andrew takes a look at the tools that are available and shares her recommendations to help you find the tool that works best for you.*

 10 min read [CSS](#), [Browsers](#) Share on [Twitter](#) or [LinkedIn](#)

MANY WEB APPLICATIONS HAVE THE REQUIREMENT OF GIVING THE user the ability to download something in PDF format. In the case of applications (such as e-commerce stores) those PDFs have to be created

Making Things Better

Anyone can contribute to specifications

You don't need to be an Invited Expert or representative.

Everyone is allowed to contribute

You don't need permission, or to be qualified in some way.

Read and **comment** on specification issues

<https://github.com/w3c/csswg-drafts/issues>

[css-grid] Ability to target items in nth-row of autofilled, implicit grid #1943

Edit

New issue

Open

rachelandrew opened this issue on 6 Nov 2017 · 14 comments



rachelandrew commented on 6 Nov 2017

Member + 😊 ...

I've had various people ask me the same question as raised here: [rachelandrew/cssgrid-ama#115](#)

The use case is as described in that issue. If the author creates the following grid:

```
nav {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
}
```

They then want to style items on the second (or nth) row differently to the first, they don't have a way to target it. I've mocked it up using MQs. <https://codepen.io/rachelandrew/pen/yPJKeo?editors=1100>

It's really a selectors issue, would the table pseudo-class selectors (<https://drafts.csswg.org/selectors-4/#table-pseudos>) work on grid items, or is this another instance of needing to be able to target a row or column of the grid, with the added complication of wanting to target the children of that row or column?

At the moment to do so, they need to go back to taking more control over the grid, rather than using auto-fill, so that they know where each item sits and can use nth-child on the items, or use media queries which makes for less flexible components.

👍 34

Assignees



No one—assign yourself

Labels



css-grid-3

Projects



None yet

Milestone



No milestone

Notifications

Customize

🔊 Unsubscribe

You're receiving notifications because you're watching this repository.



15 participants



Show use cases


As with any software development, real use cases are valuable.

Contribute **examples** and
diagrams to specifications

  rachelandrew added the **Agenda+ F2F** label on 27 Feb



css-meeting-bot commented on 28 Feb • edited by frivoal ▼

Member +  ...

The CSS Working Group just discussed `column-span`, and agreed to the following:

- RESOLVED: Clarify the spec using the notion that this is an out of flow that leaves a forced break behind
- The full IRC log of that discussion

Web Platform Tests

Not just for the web.

web- platform- tests

Navigation

[Test Suite Design](#)

[Running Tests](#)

[Writing Tests](#)

[Reviewing Tests](#)

[Project Administration](#)

Quick search

web-platform-tests documentation

The web-platform-tests project is a W3C-coordinated attempt to build a cross-browser test suite for [the Web-platform stack](#). Writing tests in a way that allows them to be run in all browsers gives browser projects confidence that they are shipping software which is compatible with other implementations, and that later implementations will be compatible with their implementations. This in turn gives Web authors/developers confidence that they can actually rely on the Web platform to deliver on the promise of working across browsers and devices without needing extra layers of abstraction to paper over the gaps left by specification editors and implementors.

The most important sources of information and activity are:

- github.com/web-platform-tests/wpt: the canonical location of the project's source code revision history and the discussion forum for changes to the code
- web-platform-tests.org: the documentation website; details how to set up the project, how to write tests, how to give and receive peer review, how to serve as an administrator, and more
- web-platform-tests.live: a public deployment of the test suite, allowing anyone to run the tests by visiting from an Internet-enabled browser of their choice
- wpt.fyi: an archive of test results collected from an array of web browsers on a regular basis
- [Real-time chat room](#): the [IRC](#) chat room named `#testing` on irc.w3.org; includes participants located around the world, but busiest during the European working day; [all discussion is archived here](#)
- [Mailing list](#): a public and low-traffic discussion list

If you'd like clarification about anything, don't hesitate to ask in the chat room or on the mailing list.

Watch a Talk

If you prefer watching a video, here is a talk introducing web-platform-tests:

Many people who work on
CSS started by **writing tests.**

Raising **print-related bugs** with browsers and print UAs

Funding people to work on
print-related specifications,
testing, and implementations

Any questions?