Code of Conduct

https://www.w3.org/Consortium/cepc/
How to raise your hand for questions
About the survey

- [https://stateofcss.com/](https://stateofcss.com/) open now
- Community effort, 3rd year running
- ~4,400 responses so far

Sacha Greif
(he/him)

Sarah Fossheim
(they/them)

Philip Jägenstedt
(he/him)
Avoiding spoilers

The survey is open until end of October

No percentages or rankings presented today

Please encourage taking the survey, but not how to respond
Accessibility
Accessibility improvements this year

- New questions on accessibility
- Accessibility of the survey itself

This work was driven by Sarah. Thanks Sarah!
Which accessibility features do you usually implement?

- Keyboard navigation
- WCAG-compliant contrast
- Alt texts
- Semantic markup
- ARIA HTML attributes

No spoilers, wait for survey to close!
Other accessibility features you usually implement 2021
Other accessibility features you usually implement

Appropriate focus handling (programmatically moving focus, retaining focus inside custom modal dialogs, etc)
Respecting user font scaling
discussed best DOM order to ensure most usable screen reader flow with UX designers
Missing Features
What do you feel is currently missing from CSS? 2021

For each matchup, pick the feature you'd most like to be able to use in CSS today.

<table>
<thead>
<tr>
<th>Container Queries</th>
<th>Being able to write styles based on the parent container's dimensions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Selector</td>
<td>The ability to target a parent element based on its children.</td>
</tr>
<tr>
<td>Nesting</td>
<td>Being able to nest styles in native CSS.</td>
</tr>
<tr>
<td>Color Functions</td>
<td>Functions for manipulating color values.</td>
</tr>
<tr>
<td>Subgrid</td>
<td>Nesting deeper-level subgrids into a parent grid.</td>
</tr>
<tr>
<td>Browser Support</td>
<td>Better browser support for existing features.</td>
</tr>
<tr>
<td>Mixins</td>
<td>Group and reuse definitions dynamically.</td>
</tr>
<tr>
<td>Scoping</td>
<td>The ability to be more specific about where your styles are applied.</td>
</tr>
</tbody>
</table>
Other Missing Features
I NEED THE PARENT SELECTOR COMBINATOR CLASS!
Other Missing Features

Backward-looking sibling selectors
Other Missing Features

Being able to style the lines in a grid
Browser Incompatibilities
Are there any CSS features you have difficulties using because of differences between browsers?
Are there any CSS features you have difficulties using because of differences between browsers?

No subgrid support outside of Firefox can make some grid layouts hard to achieve.
Subgrid (MDN)

See Chromium and WebKit tracking bugs.
Are there any CSS features you have difficulties using because of differences between browsers?

backdrop-filter: blur() [Firefox not supported]
backdrop-filter (MDN)

See Gecko tracking bug.
Are there any CSS features you have difficulties using because of differences between browsers?

the :focus-visible selector!
:focus-visible (MDN)

Shipped in Chrome/Firefox, not yet Safari.

Igalia implemented :focus-visible in WebKit, but:

“There is still the ongoing discussion about when or how this could be enabled by default in WebKit and eventually shipped in Safari.”

See WebKit tracking bug.
Are there any CSS features you have difficulties using because of differences between browsers?

Flex gap (just need more time for that though)
Are there any CSS features you have difficulties using because of differences between browsers?

Gap in flexbox, and not being able to feature detect it
Are there any CSS features you have difficulties using because of differences between browsers?

The way gap was implemented without a flex-gap alias made it impossible for me to use it.
Flex gap (MDN)

What happened?

- `grid-gap` was renamed to `gap`, works for grid+flex
- Support in flex came long after grid
- `@supports(gap: 10px)` matched before flex support
- General problem discussed in CSSWG

Should there have been a flex-gap alias?
Pain Points
For each matchup, pick the aspect of CSS you struggle with the most.

**Browser Compatibility**
Differences between Chrome, Safari, Firefox, etc.

**Interactions**
Responding to user input or other events (scroll, hover, etc.).

**Architecture & Maintenance**
File organization, dead code elimination, refactoring, etc.

**Layout & Positioning**
Setting up layouts and making sure elements go where you want them to.

**Scoping & Specificity**
Dealing with the cascade, overriding styles, etc.

**Responsive Design**
Adapting layouts and designs for different form factors.

**Form Elements Styling**
Customizing how form elements look and behave.

**Performance Issues**
Dealing with scrolling jank, making animations smoother, etc.
Other CSS Pain Points 2021
CSS properties feel arbitrary, need to learn so much by heart
Part of the community mocking anyone saying they have difficulties doing X with CSS
It seems like new properties are driven more by vendor needs than developer needs.
What Next?
How these results can be used

- Prioritization signal for browser vendors
- Input for joint [Compat 2022](#) effort
- Review with CSS Working Group?
Data used for this presentation

State of CSS 2021 TPAC analysis (spreadsheet)

Raw data:

- Browser Incompatibilities
- Other CSS Pain Points
- Other accessibility features
- Other Missing Features
Discuss!
Title

text
Title

Left

Right