CSS polyfill とその未来

@Megro.css

ken7253

Frontend developer

ブラウザの標準化まわりを追うのが趣味

最近はReactを使ったアプリケーションを書いています。

ユーザーインターフェイスやブラウザが好き。

https://github.com/ken7253

https://zenn.dev/ken7253

https://bsky.app/profile/ken7253.bsky.social

https://dairoku-studio.com

CSSの新機能 プロダクションで 使えていますか？


Baseline

Baseline Widely available

- `:is()` / `:where()`
- `width: fit-content / min-content;`
- `@layer`
- `display: block flex;` (display Multi-keyword values)
- `clamp()` / `min()` / `max()`


JS
TS

- 構文のトランスパイル
  - typescript / babel
- 組み込みオブジェクトのサポート
  - core-js / promise-polyfill
- 利用できないAPIの制限
  - @eslint/compat / typescript(lib option)

- 構文のトランスパイル
  - post-css plugins / Sass
- プロパティのサポート
  - ❌ => polyfillが必要
- 利用できないAPIの制限
  - stylelint-browser-compat

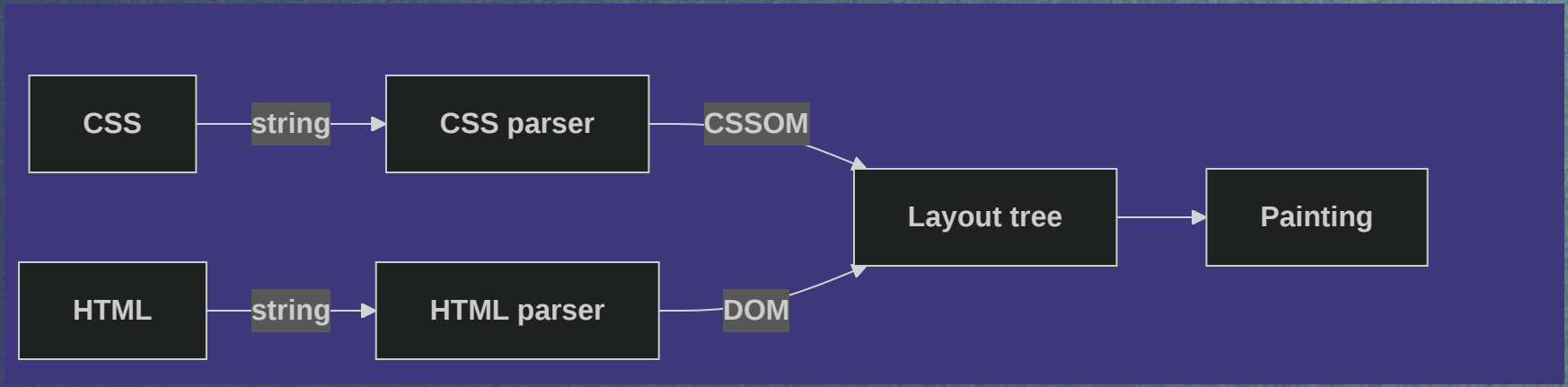
なぜCSSのPolyfillは実現が難しいのか

なぜCSSのPolyfillは実現が難しいのか

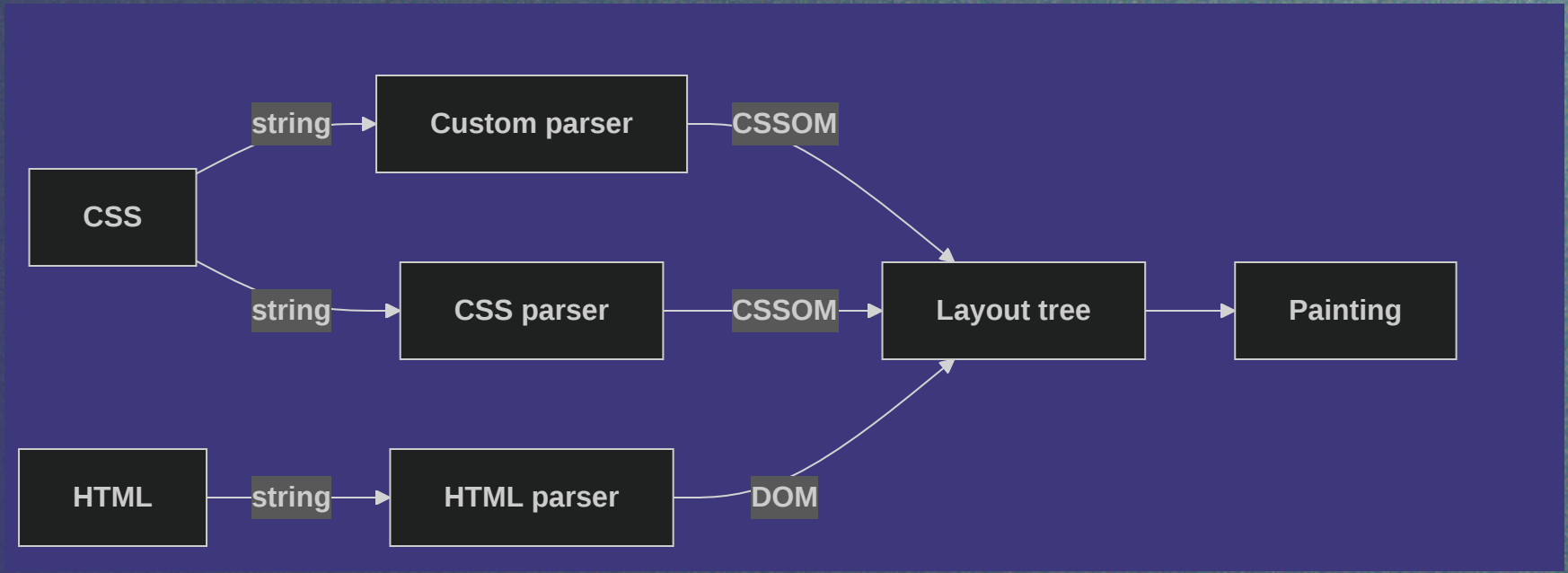
`new-feater`という新しいプロパティのpolyfillを作ることを考えます。

```
:root {
  display: block flex;
  new-feater: inherit;
}
```

なぜCSSのPolyfillは実現が難しいのか

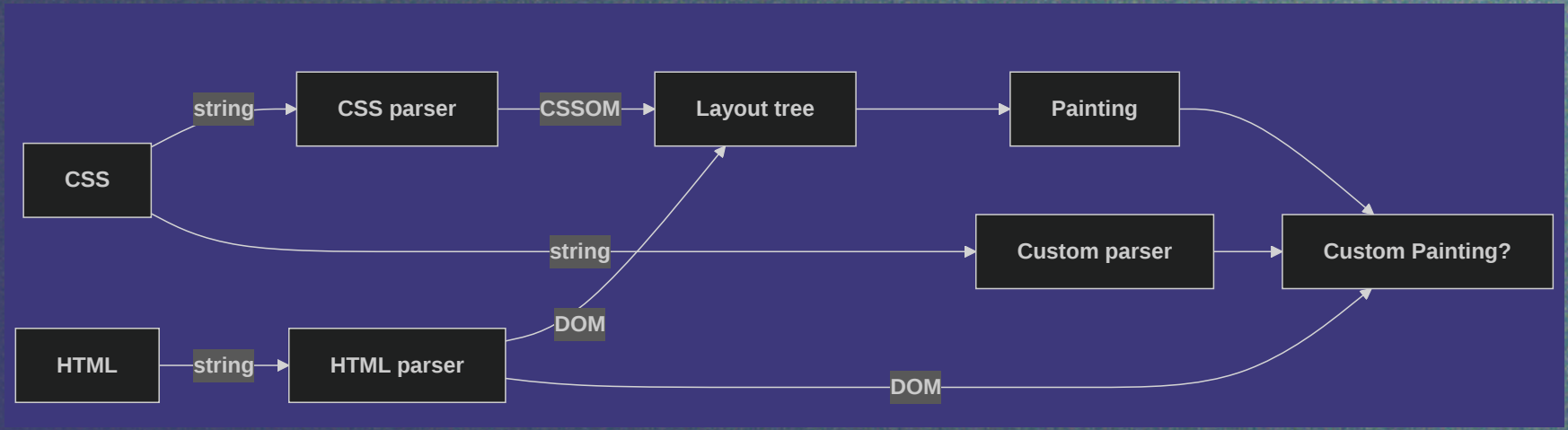


なぜCSSのPolyfillは実現が難しいのか



- 特定の`keyword`や`selector`/一部の`at-rule`の拡張が可能になりそう。
- ただこれだけではLayoutなどには関与できない。

なぜCSSのPolyfillは実現が難しいのか



Brian Kardell

Betterifying the Web

- Developer Advocate at Igalia
- Original Co-author/Co-signer of The Extensible Web Manifesto
- Co-Founder/Chair, W3C Extensible Web CG
- Member, W3C (OpenJS Foundation)
- Co-author of HitchJS
- Blogger
- Art, Science & History Lover
- Standards Geek

Follow Me On...

- Wordpress
- Medium
- Twitter
- Github
- Mastodon
- Bluesky
- Codepen
- LinkedIn
- Instagram (for art)
- Fine Art America (art for sale)

Posted on 05/02/2025

Houdini Re-Revisted

Recent presentations at BlinkOn strike some familliar notes. Seems a common theme, ideas come back.

Since I joined Igalia in 2019, I don't think I've missed a BlinkOn. This year, however, there was a conflict with the W3C AC meetings and we felt that it was more useful that I attend those, since Igalia already had a sizable contingent at BlinkOn itself and my Web History talk with Chris Lilley was pre-recorded.

When I returned, and videos of the event began landing, I was keen to see what people talked about. There were lots of interesting talks, but one jumped out at me right away: Bramus gave one called "CSS Parser Extensions" - which I wasn't familliar with, so was keen to see. Turns out it was just very beginnings of him exploring ideas to make CSS polyfillable.

This talk made me sit up and pay attention because, actually, it's really how I came to be involved in standards. It's the thing that started a lot of the conversations that eventually became the Extensible Web Community Group and the Extensible Web Manifesto, and ultimately Houdini, a joint Task Force of the W3C TAG and CSS Working Group (in fact, I am also the one who proposed the name ✨). In his talk, he hit on many of the same notes that led me there too.

Polyfills are really interesting when you step back and look at them. They can be used to make the standards development, feedback and rollout so much better. But CSS is almost historically hostile to that approach becauase it just throws away anything it doesn't understand. That means if you want to polyfill something you've got to re-implement lots of stuff that the browser already does: You've got to re-fetch the stylesheet (if you can!) as text, and then bring your own parser to parse it, and then... well, you still can't actually realistically implement many things.

But what if you could?

Houdini has stalled. In my mind, this mainly due to *when* it happened and *what it chose to focus on shipping first*. One of the first things that we all agreed to in the first Houdini meeting was that we expose the parser. This is true for all of the reasons Bramus discussed, and more. But that effort got hung up on the fact that there was a sense we first needed a typed OM. I'm not sure how true that really is. Other cool Houdini things were, I think, also hung up on lots of things that were being reworked at the time, and resource competition. But I think that the thing that really *killed it* was just what shipped first. It was not something that might be really useful for polyfilling, like custom functions or custom media queries or custom pseduo classes, or very ambitiously, something like custom layouts --- but custom paint. The CSS Working Group doesn't publish a lot of new "paints". There are approximately 0 named background images, for example. There's no `background-image: checkerboard;` for example. But the working group does publish lots of those other things like functions or