Why JavaScript Is Costing You Visibility (and what to use instead)

Jono Alderson
Independent technical SEO consultant

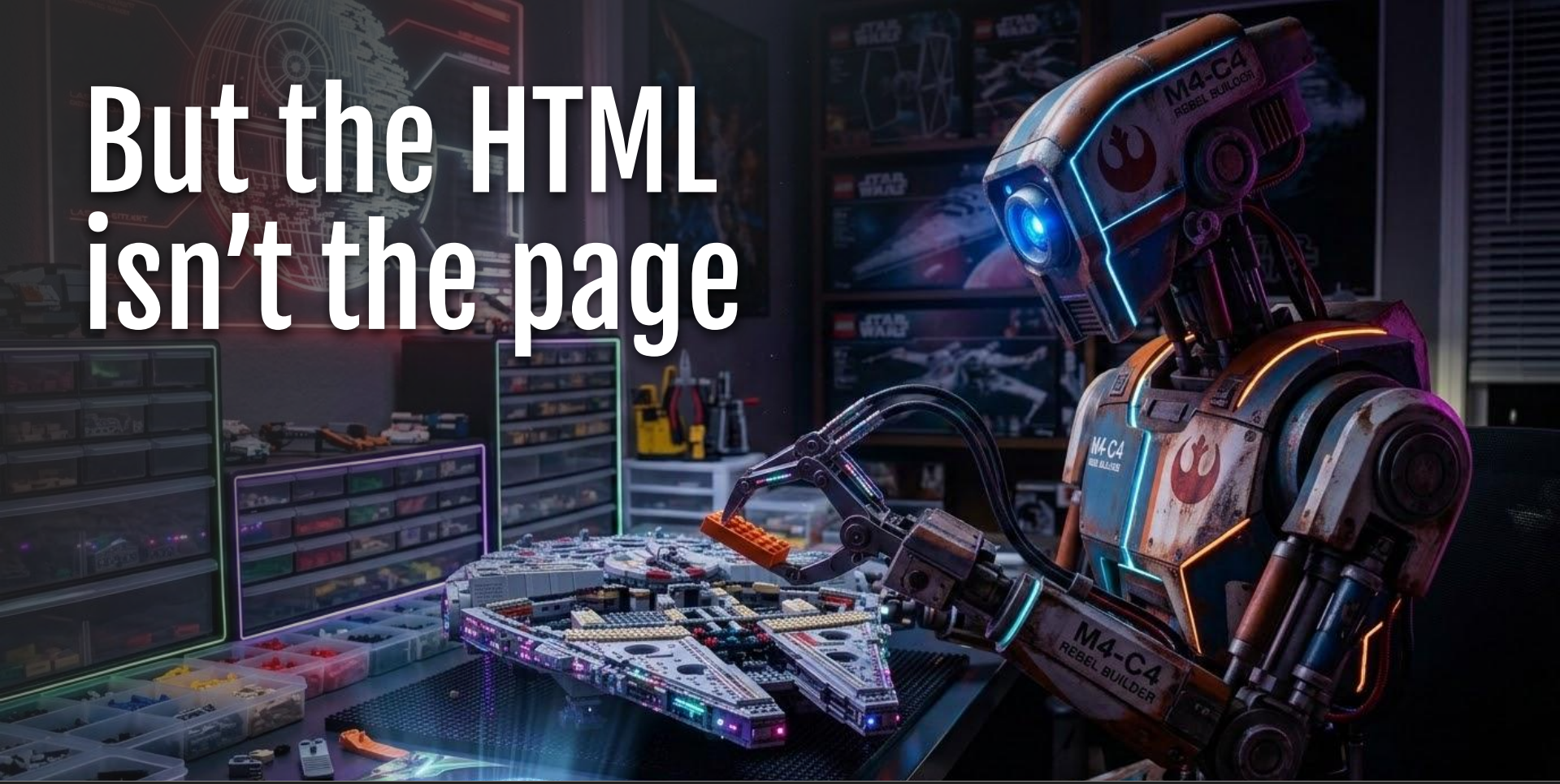
“Rendering JavaScript” isn’t a thing

Jono Alderson
Independent technical SEO consultant

A URL returns some HTML

Jono Alderson
Independent technical SEO consultant

But the HTML isn't the page



Jono Alderson
Independent technical SEO consultant

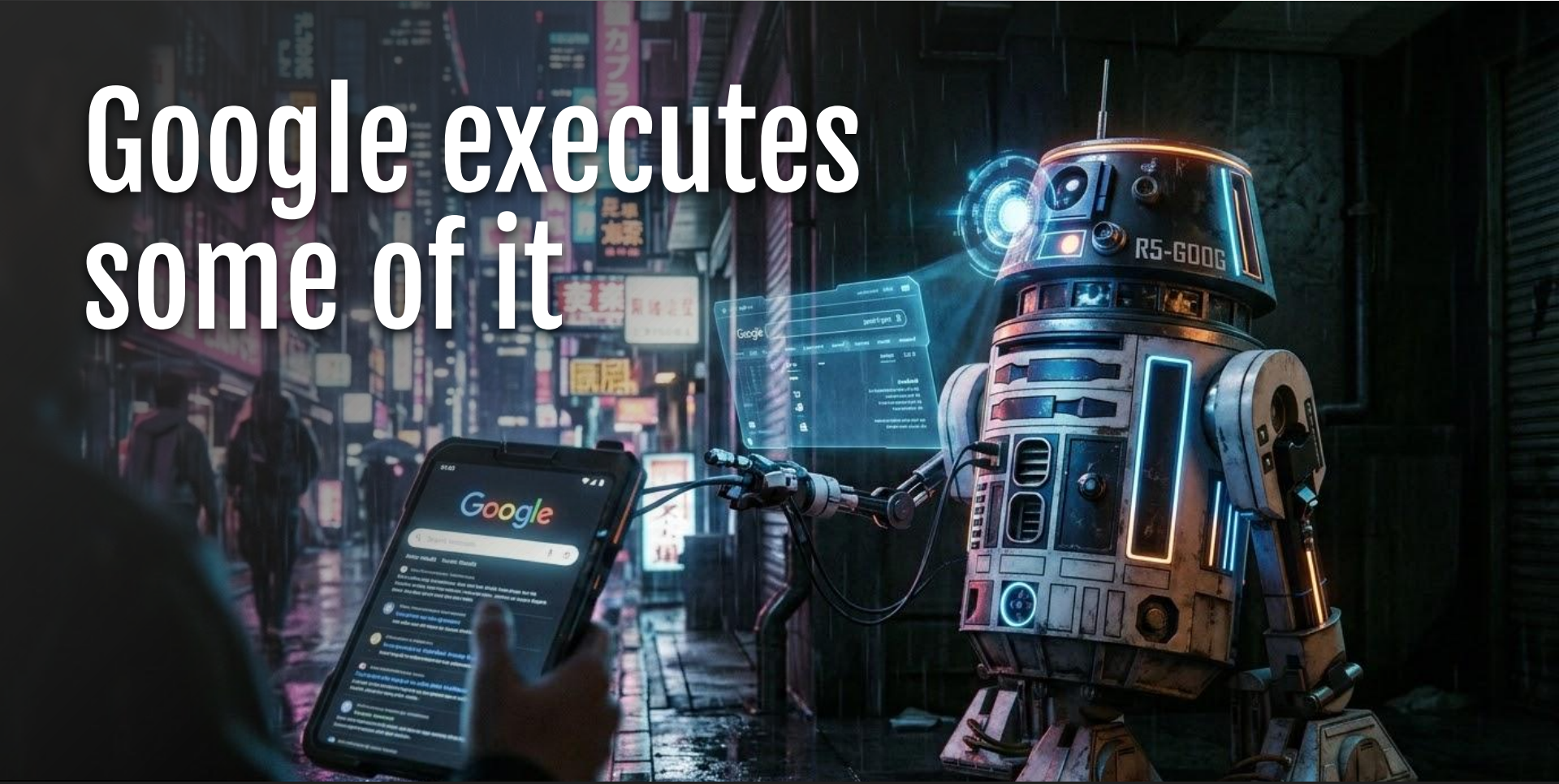
The browser constructs the page

Jono Alderson
Independent technical SEO consultant

JavaScript can rewrite anything/everything

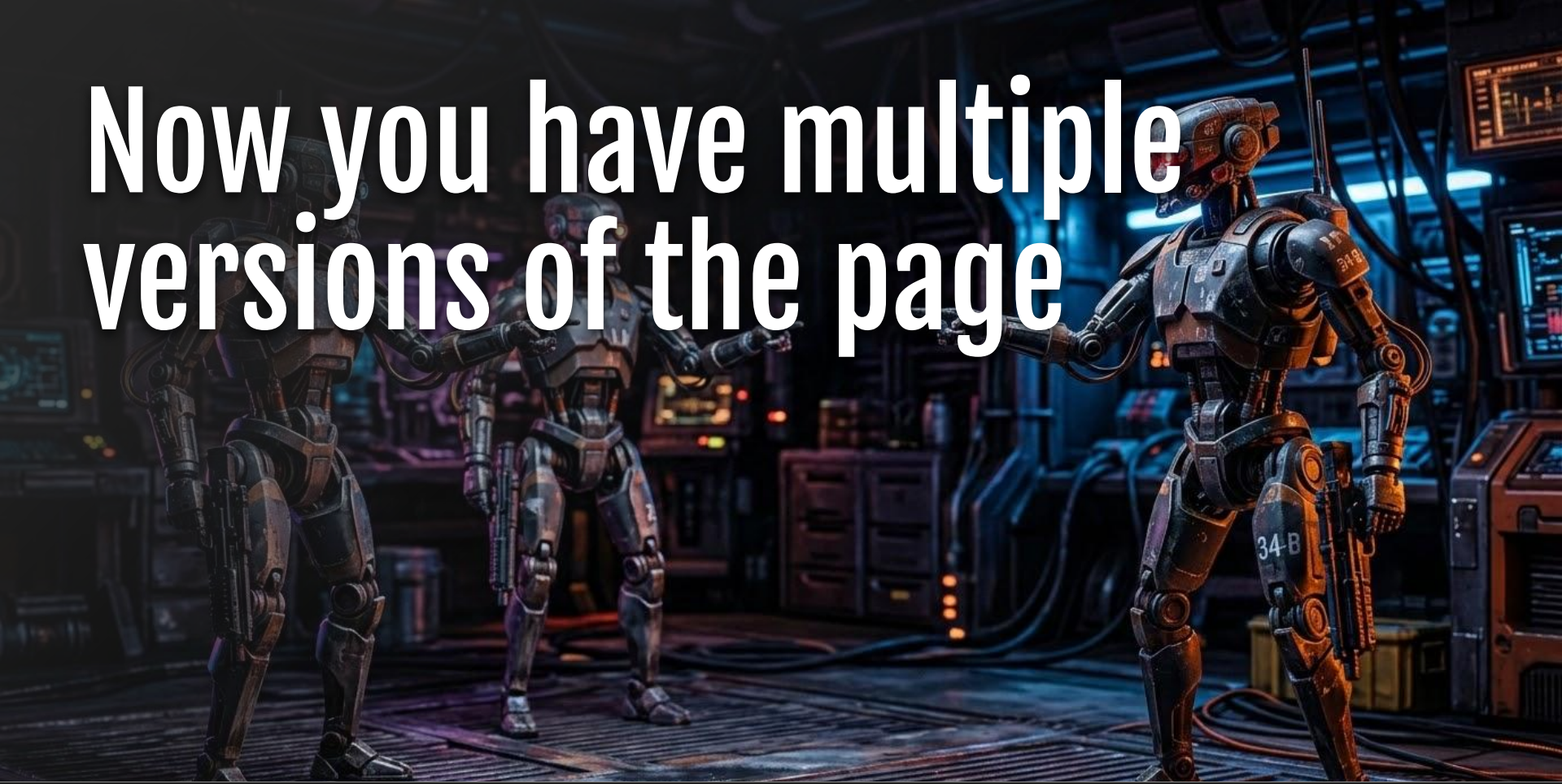
Jono Alderson
Independent technical SEO consultant

Google executes some of it



Jono Alderson
Independent technical SEO consultant

Now you have multiple versions of the page



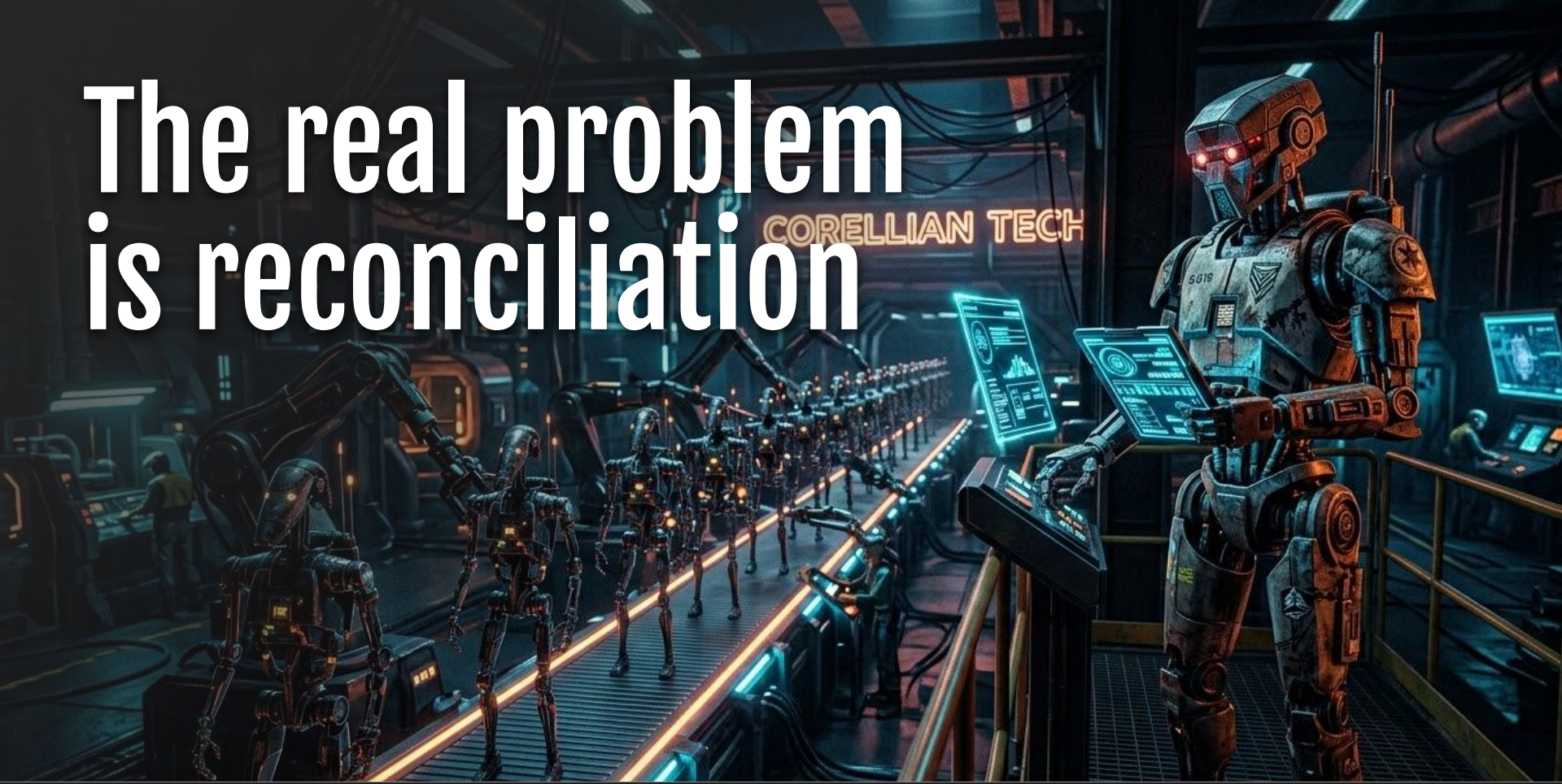
Those versions can disagree

Jono Alderson
Independent technical SEO consultant

View source, vs inspect

Jono Alderson
Independent technical SEO consultant

The real problem is reconciliation



Jono Alderson
Independent technical SEO consultant

None of this is new

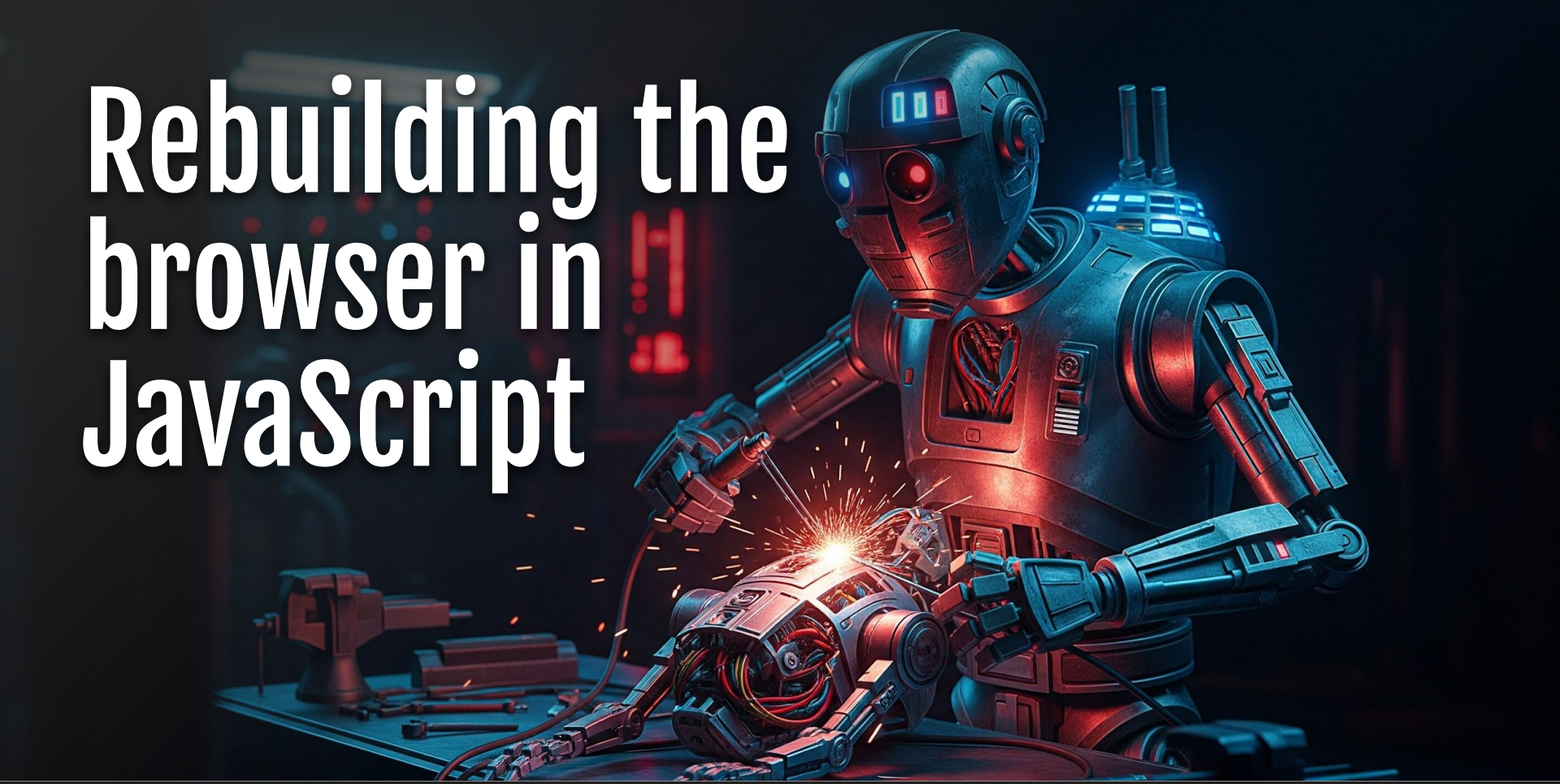
JavaScript changed jobs

Jono Alderson
Independent technical SEO consultant

Software developers started building apps

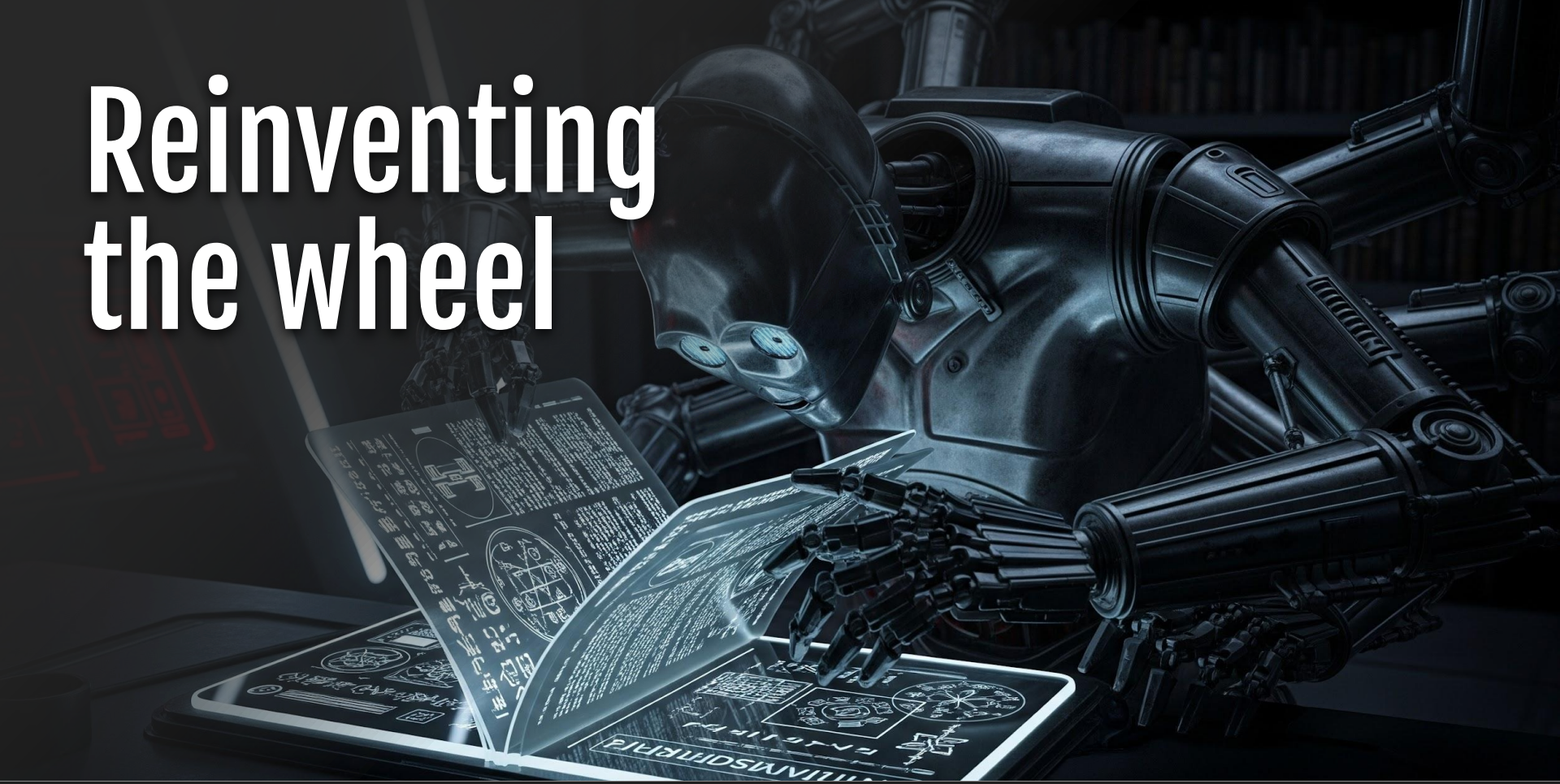
Jono Alderson
Independent technical SEO consultant

Rebuilding the browser in JavaScript



Jono Alderson
Independent technical SEO consultant

Reinventing the wheel



Jono Alderson
Independent technical SEO consultant

The infinite framework arms race

Jono Alderson
Independent technical SEO consultant

“Oh, now the back button is broken”

Jono Alderson
Independent technical SEO consultant

We created monsters

Jono Alderson
Independent technical SEO consultant

Meanwhile, the browser caught up

Jono Alderson
Independent technical SEO consultant

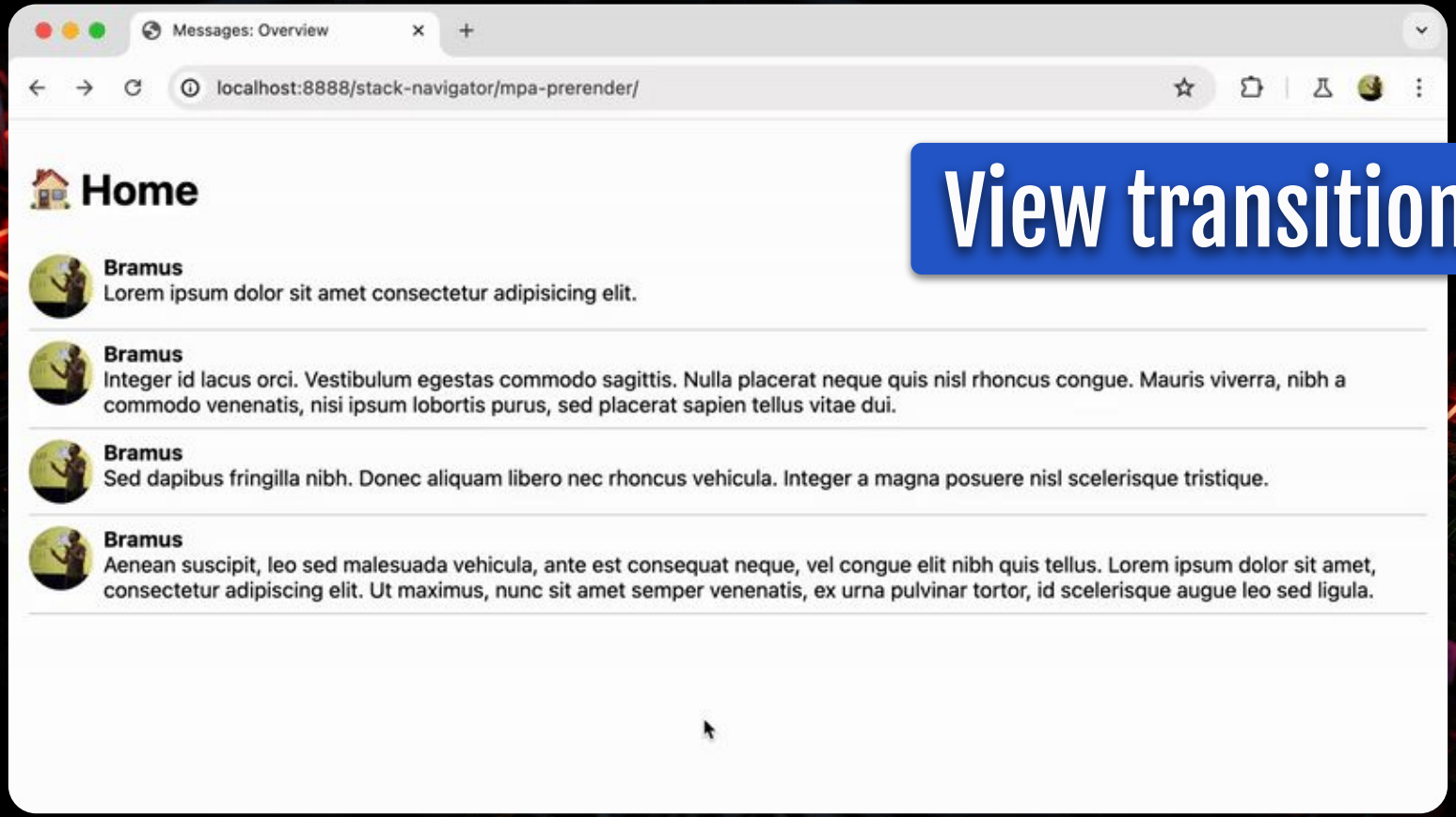
It's (unfortunately) our job to fix this

```
<section class="accordion" id="faq">
  <h3 class="acc-item">
    <button class="acc-btn" aria-expanded="false" aria-controls="a1">What is X?</button>
  </h3>
  <div id="a1" class="acc-panel" role="region" aria-hidden="true">Answer A…</div>

  <h3 class="acc-item">
    <button class="acc-btn" aria-expanded="false" aria-controls="a2">How do I Y?</button>
  </h3>
  <div id="a2" class="acc-panel" role="region" aria-hidden="true">Answer B…</div>
</section>

<style>
.acc-panel { max-height: 0; overflow: hidden; transition: max-height .25s ease; }
.acc-btn[aria-expanded="true"] + .acc-panel { max-height: 20rem; }
.acc-btn { width: 100%; text-align: left; }
</style>

<script>
(function () {
  const acc = document.querySelector('#faq');
  const buttons = acc.querySelectorAll('.acc-btn');

  function closeAll() {
    buttons.forEach(btn => {
      btn.setAttribute('aria-expanded', 'false');
      const panel = document.getElementById(btn.getAttribute('aria-controls'));
      panel.setAttribute('aria-hidden', 'true');
    });
  }

  function toggle(btn) {
    const expanded = btn.getAttribute('aria-expanded') === 'true';
    closeAll(); // one-at-a-time behavior; remove if multi-open
    btn.setAttribute('aria-expanded', String(!expanded));
    const panel = document.getElementById(btn.getAttribute('aria-controls'));
    panel.setAttribute('aria-hidden', String(expanded));
  }

  buttons.forEach(btn => {
    btn.addEventListener('click', () => toggle(btn));
    btn.addEventListener('keydown', e => {
      if (e.key === ' ' || e.key === 'Enter') { e.preventDefault(); toggle(btn); }
      if (e.key === 'ArrowDown') { e.preventDefault(); btn.closest('.acc-item').nextElementSibling?.nextElementSibling?.previousElementSibling?.querySelector('.acc-btn')?.focus(); }
      if (e.key === 'ArrowUp') { e.preventDefault(); btn.closest('.acc-item').previousElementSibling?.previousElementSibling?.querySelector('.acc-btn')?.focus(); }
    });
  });
})();
</script>
```


```
<details class="faq">
  <summary>What is X?</summary>
  <p>Answer A…</p>
</details>

<style>
summary {cursor:pointer;list-style:none}
summary::after {content:'▸';float:right;transition:.2s}
details[open] summary::after {transform:rotate(90deg)}
</style>
```

THINGS YOU SHOULD BE PLAYING WITH

Jono Alderson
Independent technical SEO consultant



View transitions

Jono Alderson
Independent technical SEO consultant

Create view transition in 1 CSS line

View transitions

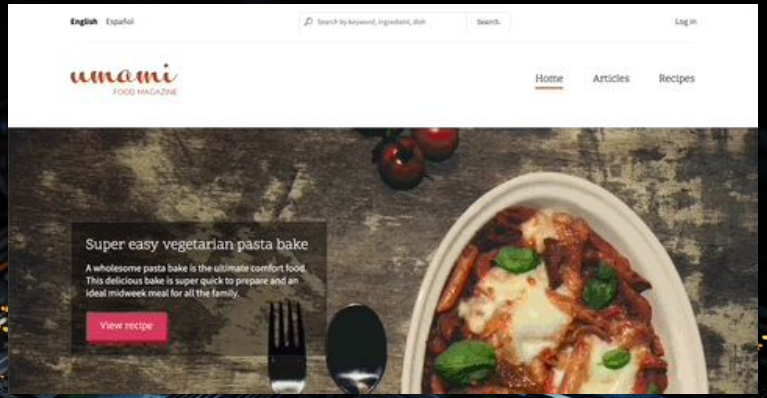
Next page

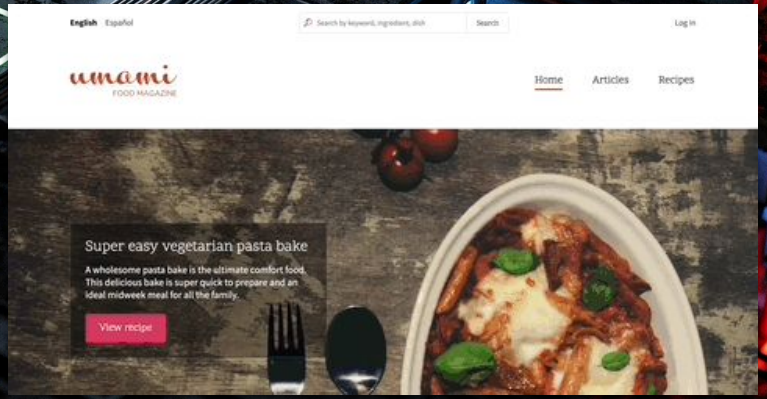
```
@view-transition {
    navigation: auto;
}
```


Theodore Soti
@TheoSoti

Checkout my blog
→ theosoti.com

Repost





API REFERENCE › COMPONENTS ›

<ViewTransition>

Canary

The `<ViewTransition />` API is currently only available in React's Canary and Experimental channels.

Learn more about React's release channels here.

`<ViewTransition>` lets you animate elements that update inside a Transition.

```
import {ViewTransition} from 'react';

<ViewTransition>
  <div>...</div>
</ViewTransition>
```


Pure CSS carousels

Horizontal Cards

Scroll Buttons | Scroll Markers | Scroll-State Queries

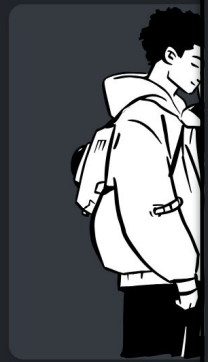
Scroll Initial Target | Anchor | Full Bleed

Buttons and markers both anchored to the bottom

Jam Strut

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

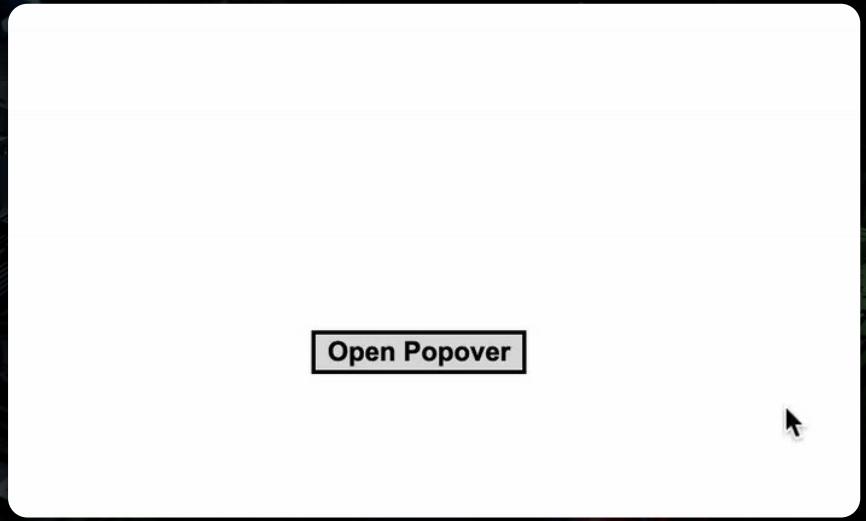
Check it out



Jono Alderson
Independent technical SEO consultant

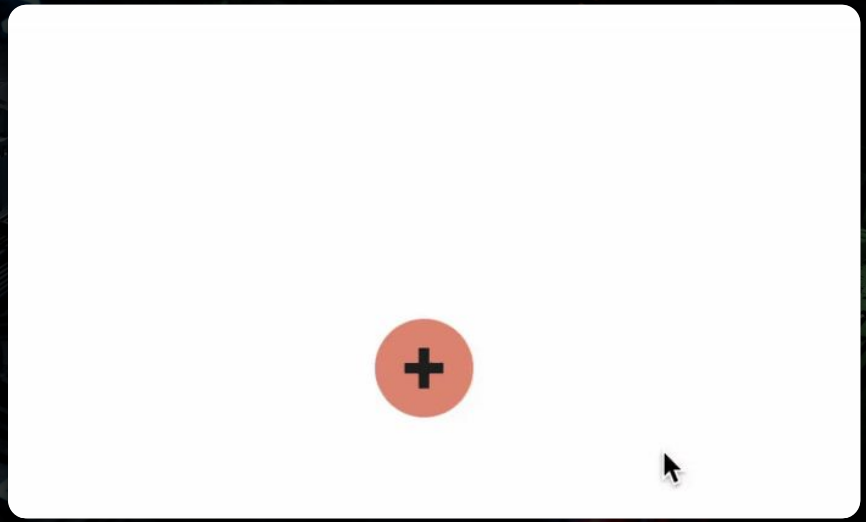
Popover

```
<button popovertarget="mypopover">
  Toggle popover
</button>

<div id="mypopover" popover>
  Popover content
</div>
```



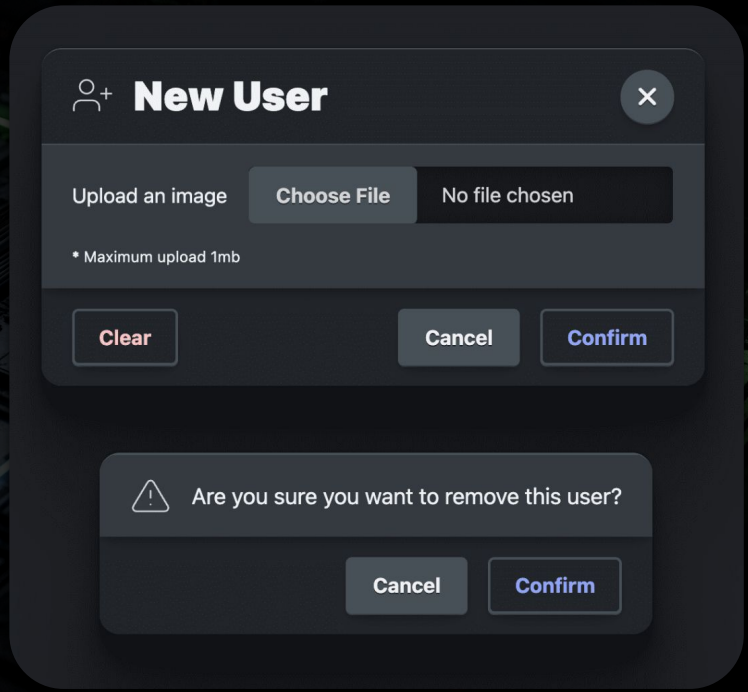
Popover

```
<button popovertarget="mypopover">
  Toggle popover
</button>

<div id="mypopover" popover>
  Popover content
</div>
```



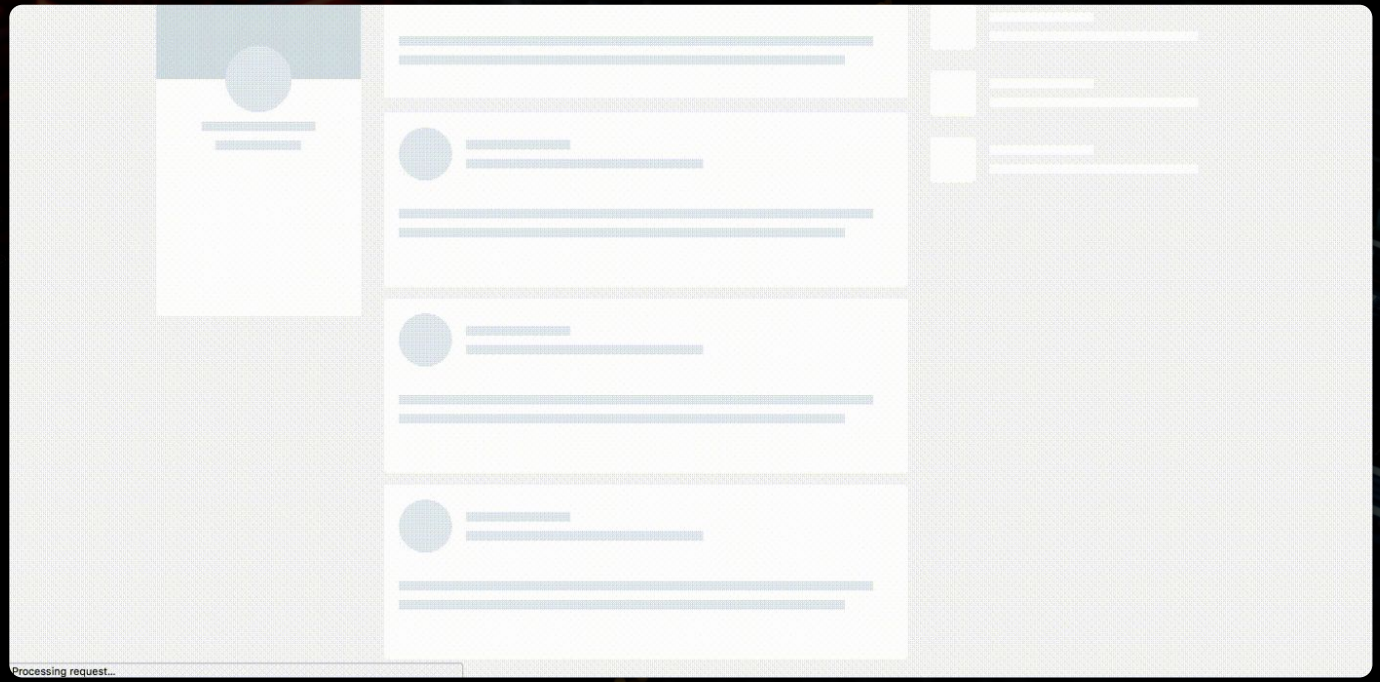
Dialog

```
<button onclick="example.showModal()">
  Open
</button>

<dialog id="example">
  <p>Hello world</p>

  <button onclick="example.close()">
    Close
  </button>
</dialog>
```



Jono Alderson
Independent technical SEO consultant

Content visibility



Jono Alderson
Independent technical SEO consultant

Content visibility

```
<ul>
  <li class="card">[...]</li>
  <li class="card">[...]</li>
  [...]
</ul>

.card {
  content-visibility: auto;
  contain-intrinsic-size: 400px 300px;
}
```


Containment

```
.card {
  contain: layout paint style;
  /* or: contain: content; */
}
```


Change management

```
.hero img {
  transition: transform .3s ease;
  will-change: transform; /* pre-promotes layer */
}

.hero img:hover {
  transform: scale(1.05);
}
```


Speculation rules

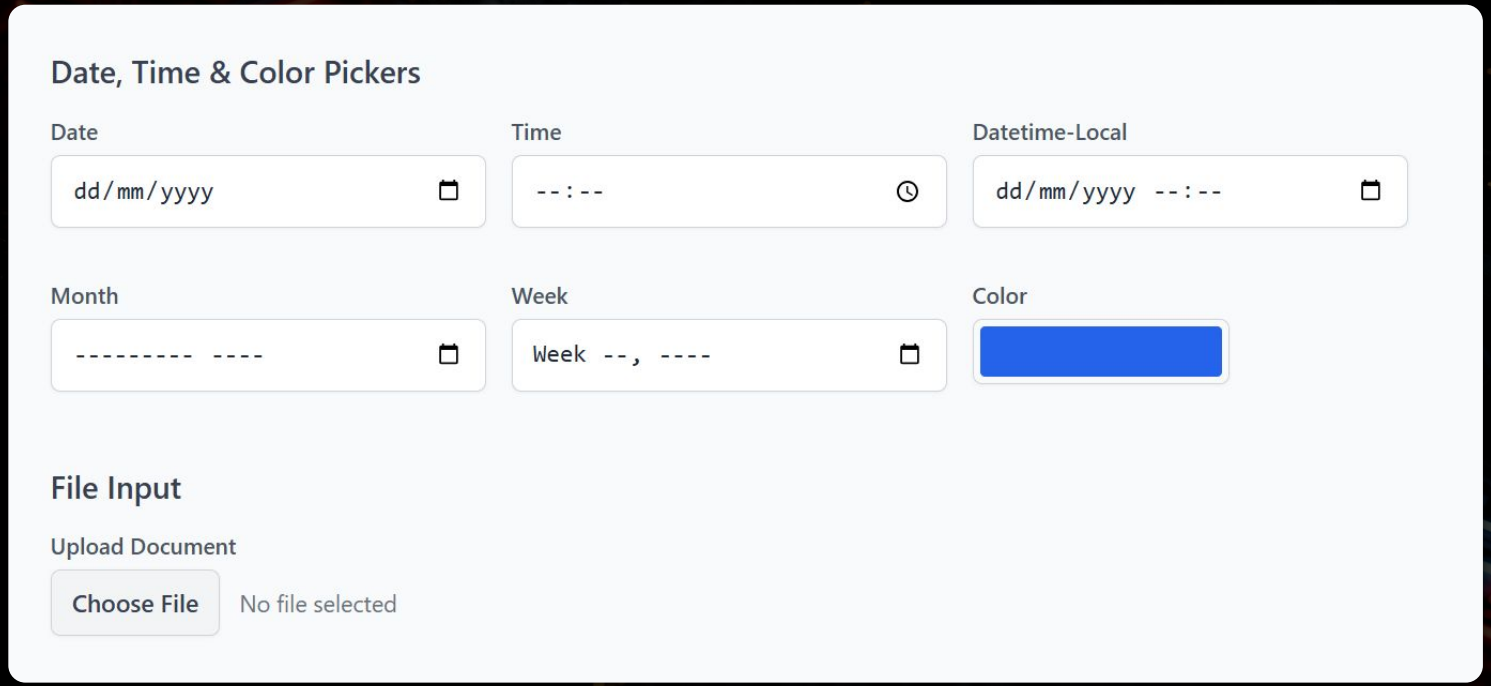
```
<script type="speculationrules">
{
  "prefetch": [
    {
      "where": { "href_matches": "^/" }, // same-origin navigations
      "eagerness": "moderate"              // hover 200ms or pointerdown
    }
  ]
}
</script>
```


Speculation rules

```
"prefetch": [
  {  // High-value links you've explicitly marked up
    "where": { "selector_matches": "a[data-prefetch]" },
    "eagerness": "immediate"
  },
  {  // Exclude sensitive routes
    "where": {
      "and": [
        { "href_matches": "^/" },
        { "not": { "href_matches": "^/(logout|checkout|admin)" } },
        { "not": { "selector_matches": "a[data-prefetch='off']" } }
      ]
    },
    "eagerness": "moderate"
  }
]
```

```
"prerender": [
  { // Only prerender links you explicitly opt-in
    "where": { "selector_matches":
"a[data-prefetch='prerender']" },
    "eagerness": "immediate"
  },
  { // Or prerender highly likely next pages
    "where": {
      "and": [
        { "href_matches": "^/article/[^#?]+$" },
        { "selector_matches": "a[rel='next']" }
      ]
    },
    "eagerness": "conservative"  // on pointer/touch down
  }
]
```


Form controls



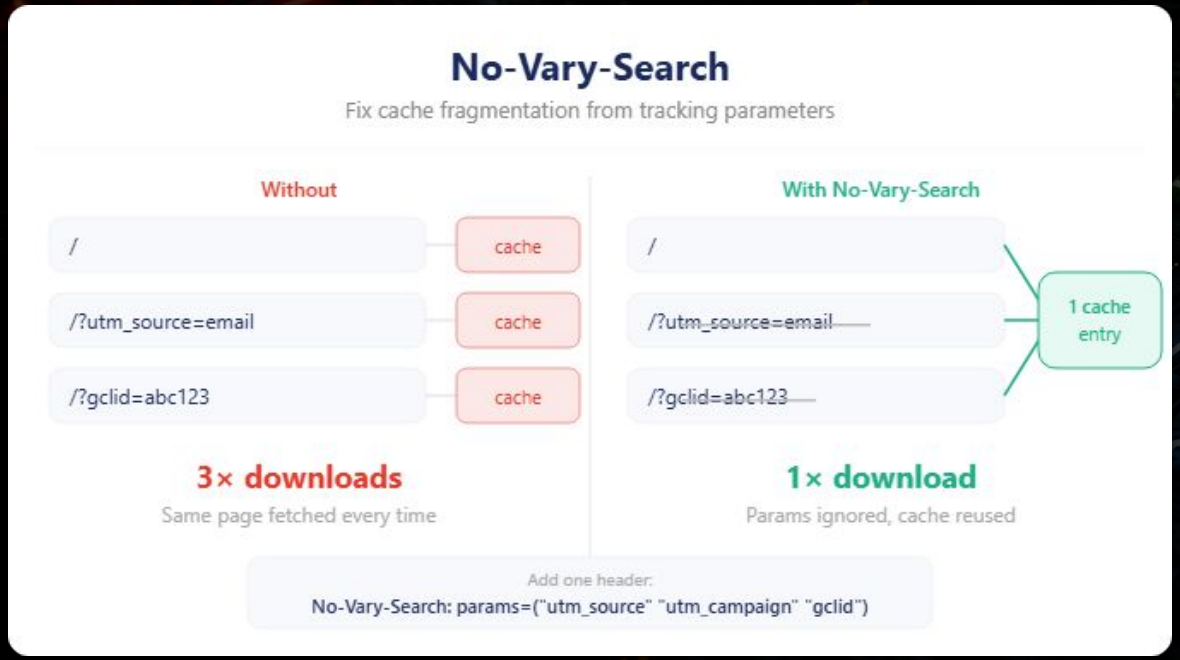
Form controls

```
form:has(:invalid) {
  outline: 1px solid red;
}

.field:has(input:invalid) label {
  color: red;
}

.field.password:has(input:focus) .strength-hint {
  opacity: 1;
}
```


Lazy loading <video>

```
<video
  src="cats.mp4"
  poster="cat.webp"
  preload="metadata"
  loading="lazy"
>
```

No-vary-search

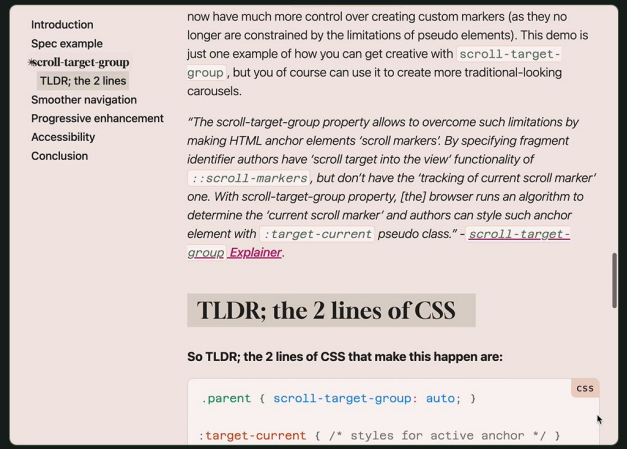


Jono Alderson
Independent technical SEO consultant

Creating a scroll-spy with 2 lines of CSS

Published on July 29, 2025

Did you know that there's a new CSS feature **landing in Chrome 140** that let's you very easily create trackable table of contents? This "scroll-spy" effect can be achieved with a new CSS property called `scroll-target-group` combined with the `:target-current` pseudo class.



now have much more control over creating custom markers (as they no longer are constrained by the limitations of pseudo elements). This demo is just one example of how you can get creative with `scroll-target-group`, but you of course can use it to create more traditional-looking carousels.

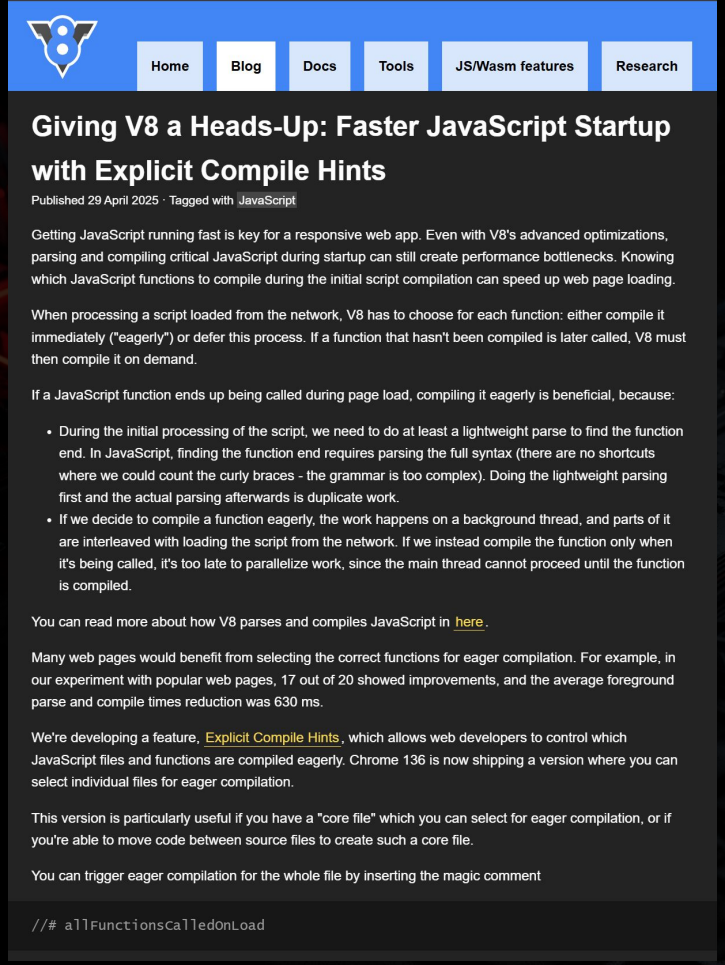
"The scroll-target-group property allows to overcome such limitations by making HTML anchor elements 'scroll markers'. By specifying fragment identifier authors have 'scroll target into the view' functionality of `::scroll-markers`, but don't have the 'tracking of current scroll marker' one. With scroll-target-group property, [the] browser runs an algorithm to determine the 'current scroll marker' and authors can style such anchor element with `:target-current` pseudo class." - scroll-target-group Explainer.

TLDR; the 2 lines of CSS

So TLDR; the 2 lines of CSS that make this happen are:

```css
.parent { scroll-target-group: auto; }

:target-current { /* styles for active anchor */ }
```


Jono Alderson
Independent technical SEO consultant



*"In our experiment with popular web pages, **17 out of 20** showed improvements, and the average foreground parse and compile times reduction was **630ms**"*

//# allFunctionsCalledOnLoad

Jono Alderson
Independent technical SEO consultant

All of this works together

Jono Alderson
Independent technical SEO consultant

Jono Alderson
Independent technical SEO consultant

Render on the server

Jono Alderson
Independent technical SEO consultant

Jono Alderson
Independent technical SEO consultant

Looking ahead

Jono Alderson
Independent technical SEO consultant

Structure = Accessibility

Jono Alderson
Independent technical SEO consultant

Machines are the new users

Jono Alderson
Independent technical SEO consultant

Browser readiness = AI readiness

Jono Alderson
Independent technical SEO consultant

Build for humans, Build for machines

Jono Alderson
Independent technical SEO consultant

Why JavaScript Is Costing You Visibility

(and what to use instead)

Jono Alderson
Independent technical SEO consultant