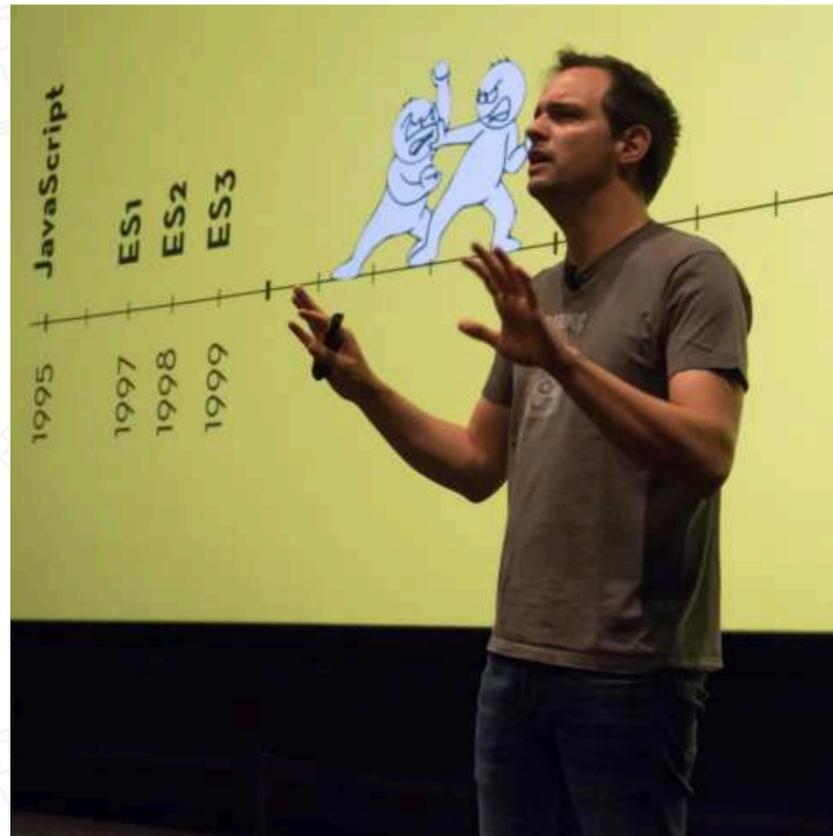


Embrace The Platform

@bramus

2022.03.03 — UX Ghent

Bramus



[@bramus](https://www.bram.us/)

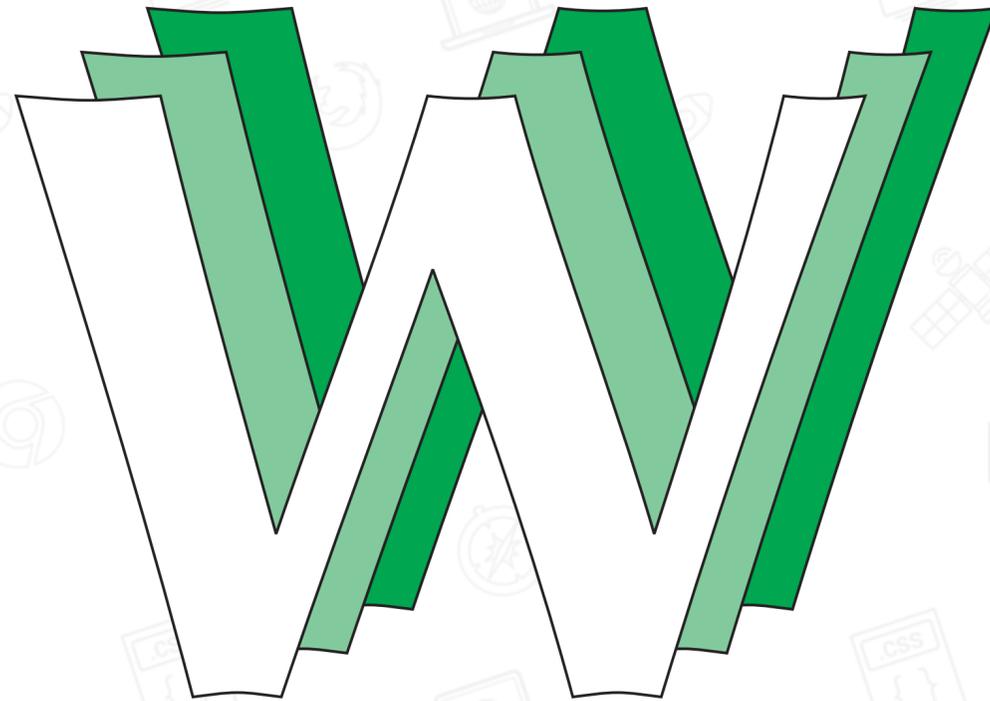


<https://www.bram.us/>
[@bramusblog](https://www.bram.us/@bramusblog)

WWW



Let's Share What We Know



World Wide Web





1989 | Tim Berners-Lee

Vague but exciting ...

CERN DD/OC
Information Management: A Proposal

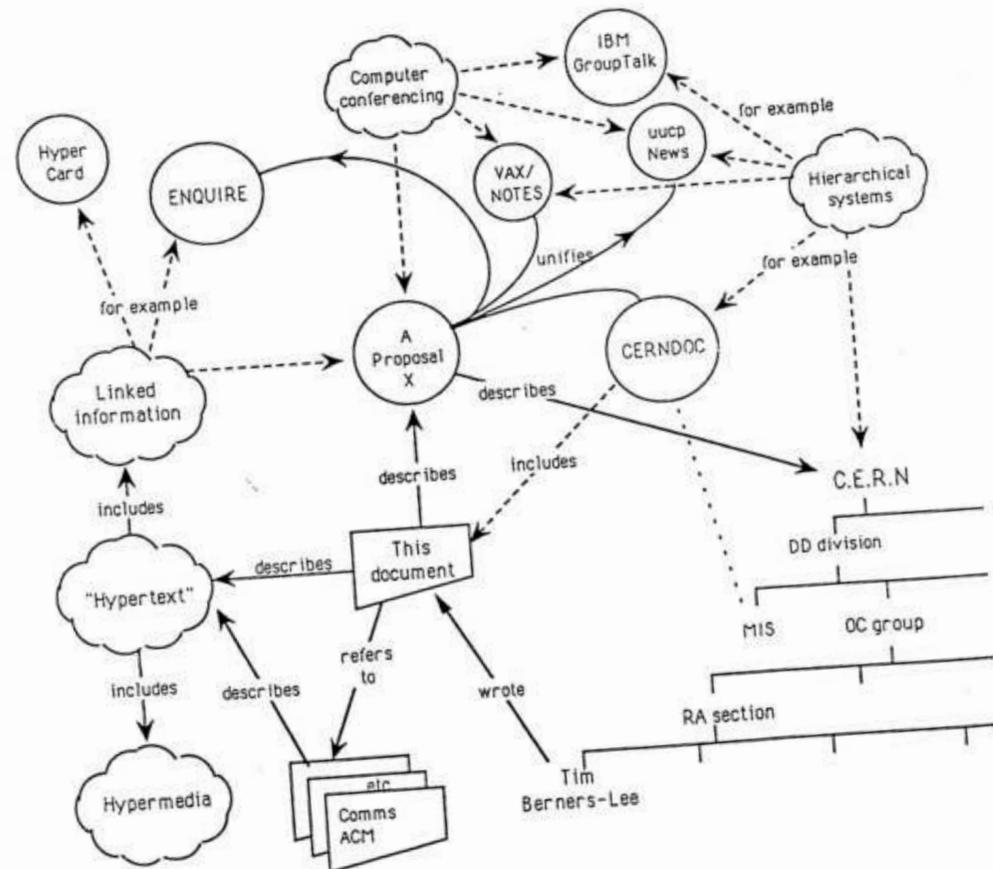
Tim Berners-Lee, CERN/DD
March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control



1989 | Tim Berners-Lee

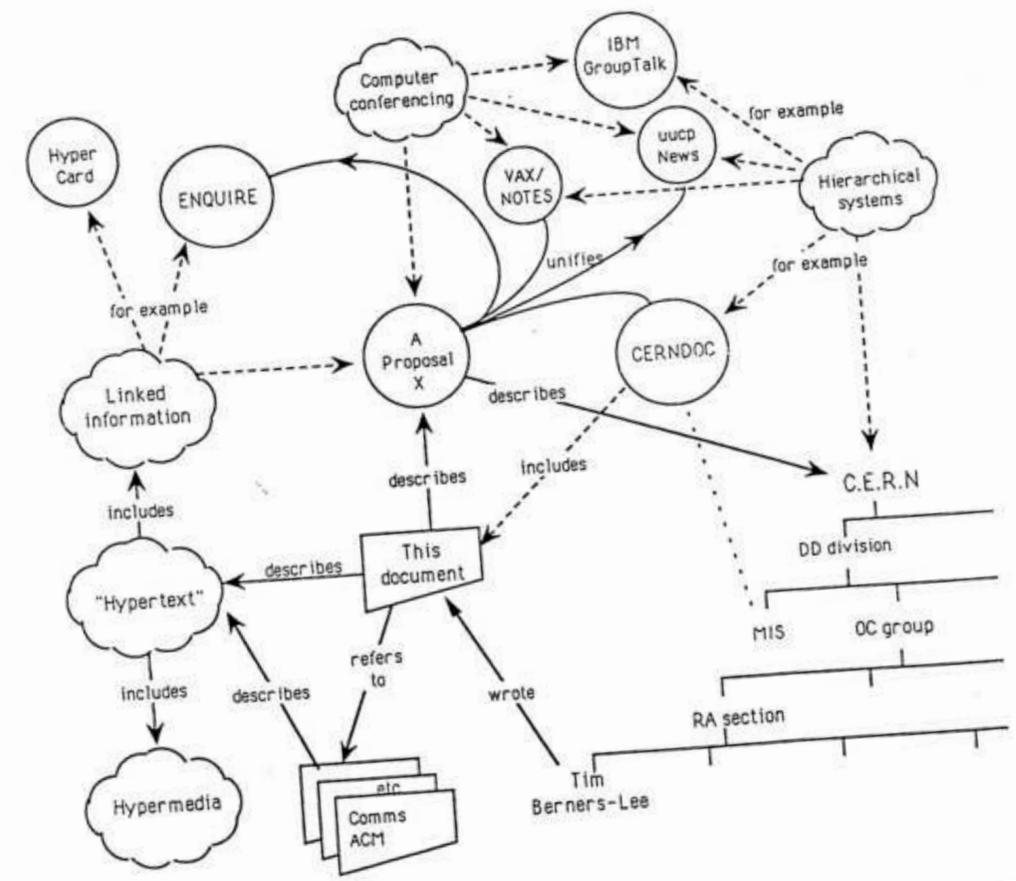
Vague but exciting...

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control



1989 | Tim Berners-Lee

“Vague, but exciting ...”

TBL's Boss

<https://www.bram.us/>

Vague but exciting ...

CERN DD/OC
Information Management: A Proposal

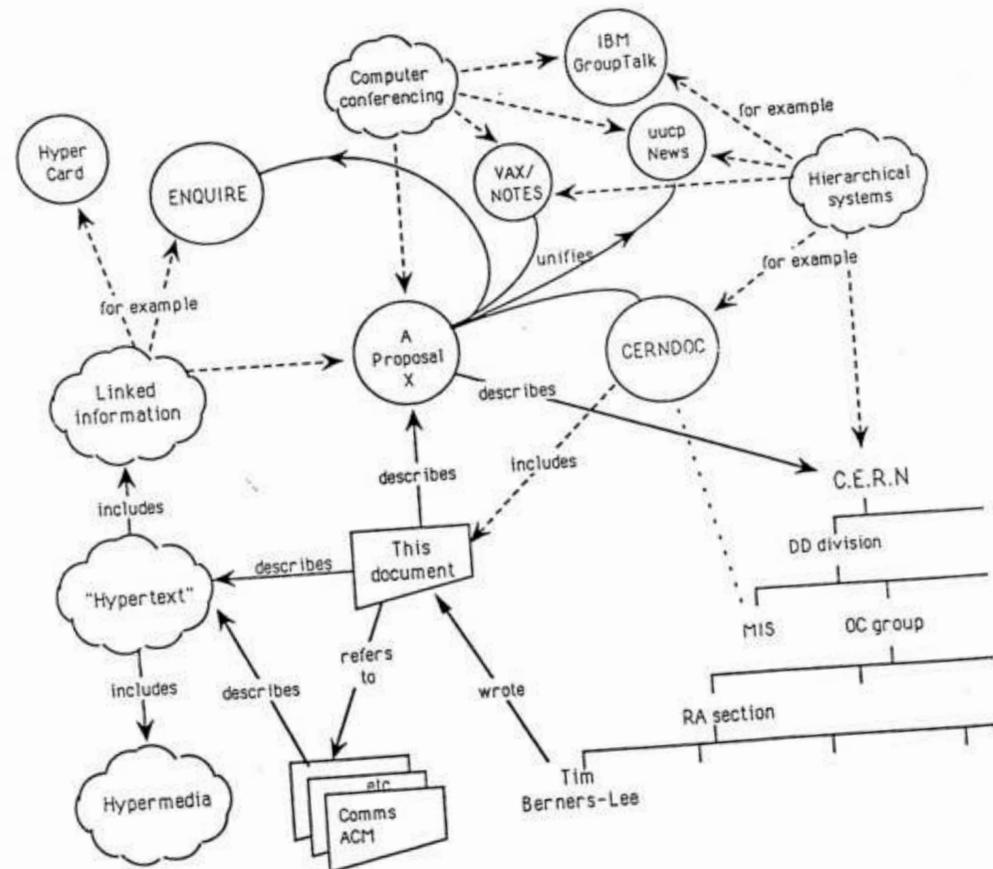
Tim Berners-Lee, CERN/DD
March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control



1989 | Tim Berners-Lee



1998 | Fifa World Cup

Web Architecture from 50,000 feet

<https://www.w3.org/DesignIssues/Architecture.html>

An attempt to give a high-level overview of the architecture of the WWW

<https://www.w3.org/DesignIssues/Architecture.html>

Web Architecture from 50,000 feet

Topics

- Principles of Design
- URI Schemes
- The Universal Web
- Web Services
- Format Negotiation
- Human Readable Information
- The Semantic Web

Web Architecture from 50,000 feet

Topics

- **Principles of Design**
- URI Schemes
- The Universal Web
- Web Services
- Format Negotiation
- **Human Readable Information**
- The Semantic Web

Human Readable Information

Style Sheets

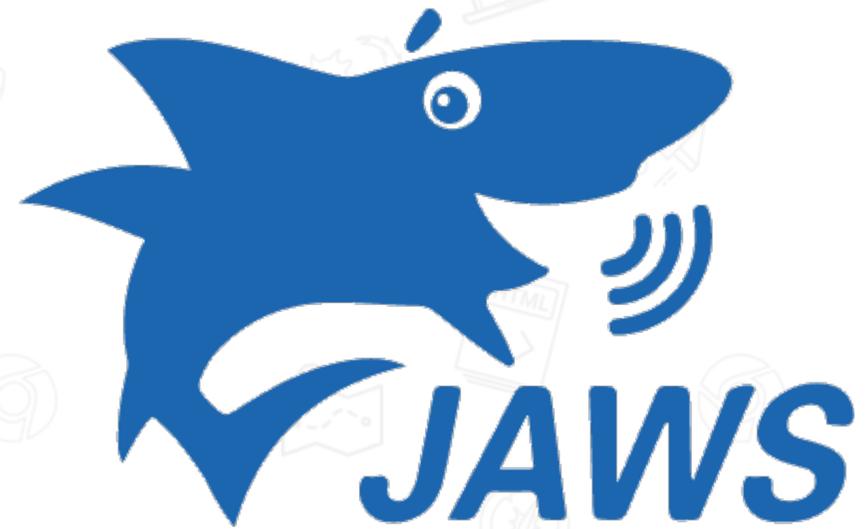
Style essentially is the **mapping between the abstract content of a document and the physical form in which it is displayed, spoken, performed or in general presented, to its recipient.**

Human Readable Information

Separation of Form and Content

It is an **essential part of Web architecture**, making possible the **independence of device [...]**

The addition of presentation information to HTML when it could be put into a style sheet breaks this rule.



Principles of Design

Rule of Least Power

Choose the least powerful language
suitable for a given purpose

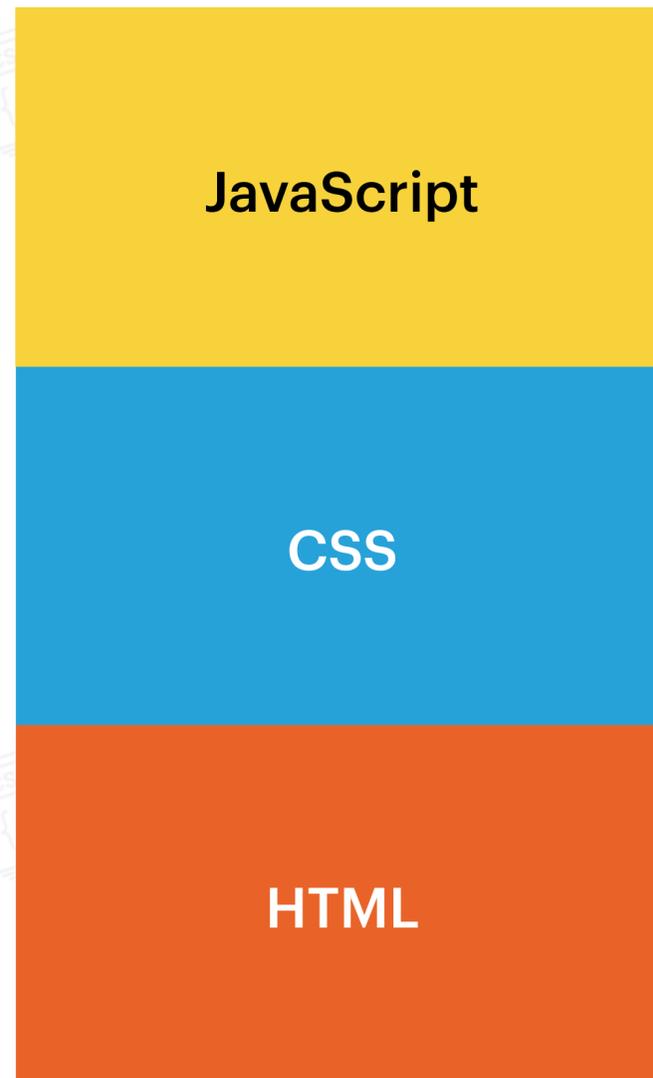
The Rule of Least Power, on the Web

HTML

CSS

JavaScript

The Rule of Least Power, on the Web



The Rule of Least Power, on the Web

 Derek Featherstone

In the web front-end stack — HTML, CSS, JS, and ARIA — if you can solve a problem with a simpler solution lower in the stack, you should. It's less fragile, more foolproof, and just works

<http://simplyaccessible.com/article/data-attributes/>

The Broken Web

The Broken Web

 Drew Devault

My browser has been perfectly competent at submitting HTML forms for the past 28 years, but for some stupid reason some developer decided to reimplement the form in JavaScript, and now I can't pay my electricity bill without opening up the dev tools

<https://drewdevault.com/2021/10/17/Reliability.html>

```
✘ $(document).ready(function () {
    $("a").on("click", function (event) {
        if (this.hash !== "") {
            event.preventDefault();

            var hash = this.hash;

            $("html, body").animate({
                scrollTop: $(hash).offset().top,
            }, 800, function() {
                window.location.hash = hash;
            });
        }
    });
});
```

```
✓ html {  
    scroll-behavior: smooth;  
}
```

The Rule of Least Power, on the Web



The Rule of Least Power, on the Web

Examples

- **Smooth Scrolling**
 - 👉 No need for JavaScript as CSS can do that
- **Communicate errors from your JSON-based API?**
 - 👉 Don't use an HTTP 200 with `{error: true}` in the response body, but an HTTP Status Code to communicate the error instead
- **Closing a `<dialog>` via JavaScript?**
 - 👉 A `<form>` element with `[method="dialog"]` will do just fine

The Rule of Least Power, on the Web

Examples

- **Want to lazy load images?**
 - 👉 That will soon be supported by all modern browsers directly in HTML markup with the [loading] attribute
- **A customizable <select>?**
 - 👉 That's in development as we speak
- **Need to prevent certain characters in form inputs?**
 - 👉 Don't disable pasting but use a proper input type + [pattern] attribute

The Rule of Least Power, on the Web

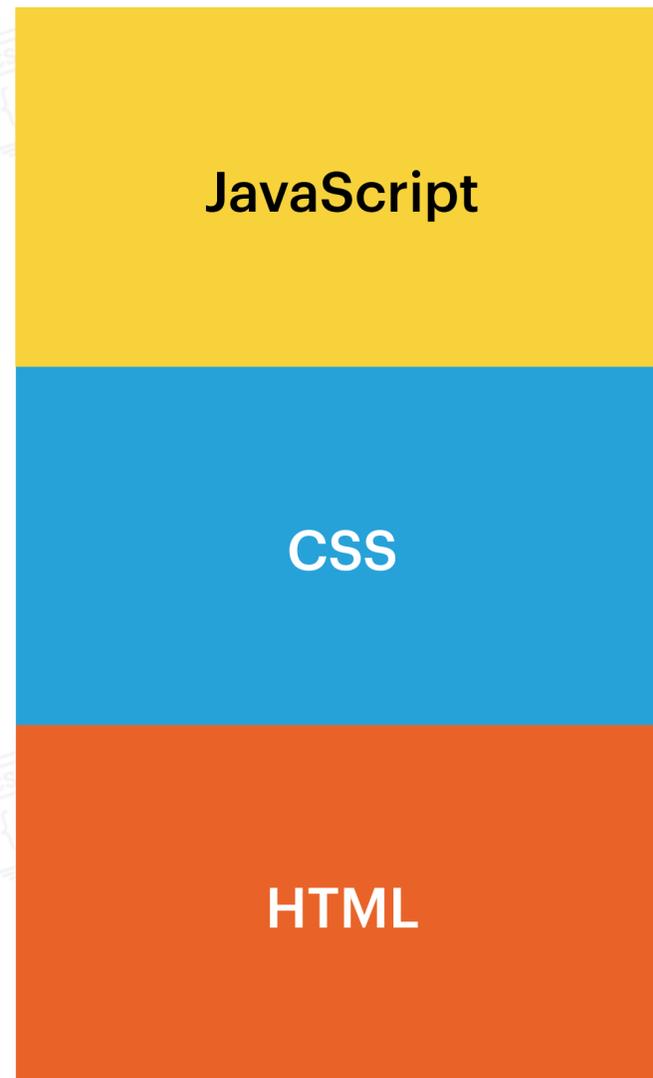
Examples

- **Need collapsing sections?**
👉 `<details>` and `<summary>` are your friends
- **Container Queries?**
👉 Soon, no longer a dream

•

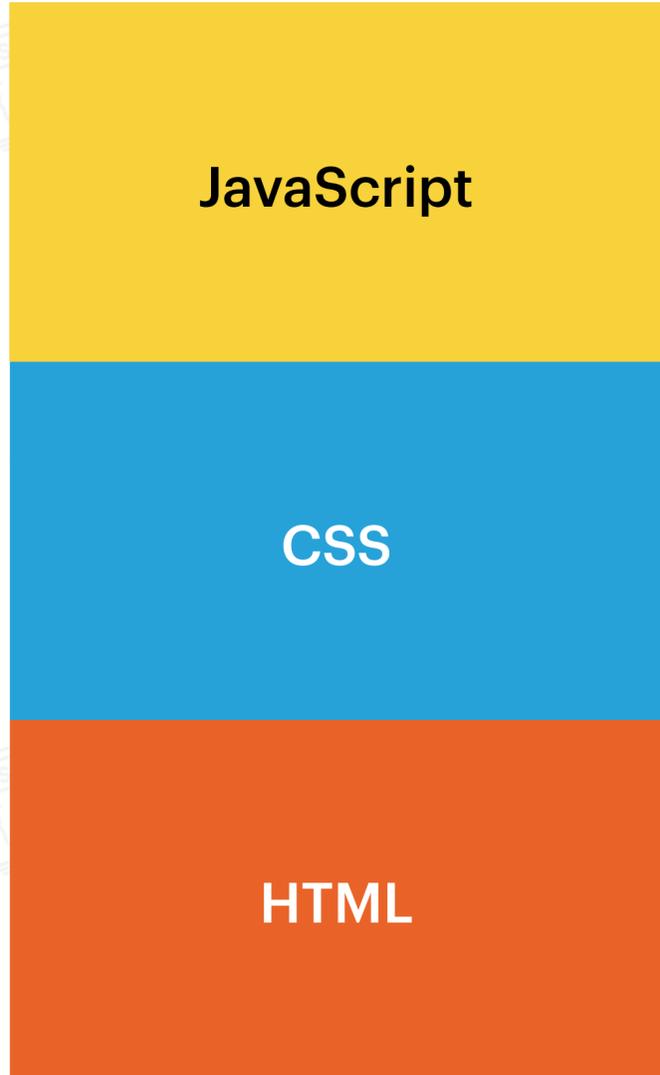
...

The Rule of Least Power, on the Web

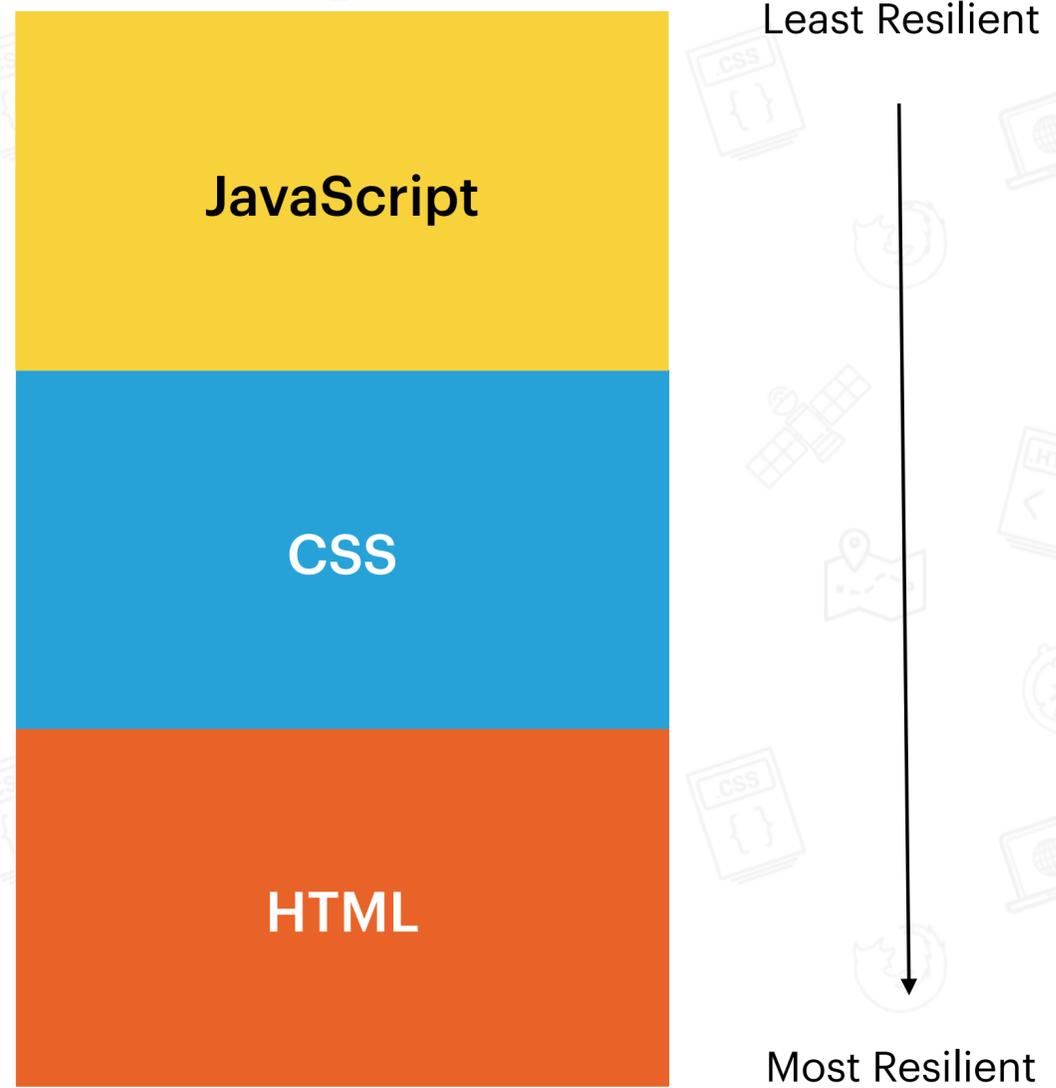


Resilience

The Rule of Least Power, on the Web



The Rule of Least Power, on the Web



The Rule of Least Power, on the Web

 Derek Featherstone

In the web front-end stack — HTML, CSS, JS, and ARIA — if you can solve a problem with a simpler solution lower in the stack, you should.

It's less fragile, more foolproof, and just works

<http://simplyaccessible.com/article/data-attributes/>

Bramus Van Damme (@bramus) | instagram.com/bramus/

Elements Console Sources Network

top Filter Default levels 1 Issue: 1 1 hidden

Uncaught Error: Requiring unknown module "73". (index):287

- at u ((index):287:1572)
- at c ((index):287:1151)
- at i ((index):287:895)
- at r ((index):287:390)
- at 4c362d5c3b6d.js:1227:1



Notifications / Twitter

twitter.com/notifications

Notifications

All Mentions

Jordi Sánchez liked 2 Tweets you were mentioned in

If these are standalone functions, I'd be more explicit about what they return:

```
getHighestSpecificity/getLowestSpecificity
```

Or combine them:

```
getSpecificity(specificity: 'highest' | 'lowest')
```

If it's an array of numeric specificities, you could also just do `Math.max/min`.

[Show all](#)

Mladen Panić and 5 others followed you

Stephan and 115 others liked your Tweet

TIL: You can inspect system web views in macOS.

blog.jim-nielsen.com/2022/inspectin...

Hat tip, @jimniels! pic.twitter.com/xkpmhTwAgP

Elements Console Sources Network

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

```
.css-1dbjc4n {
  -ms-flex-align: stretch;
  -ms-flex-direction: column;
  -ms-flex-negative: 0;
  -ms-flex-preferred-size: auto;
  -webkit-align-items: stretch;
  -webkit-box-align: stretch;
  -webkit-box-direction: normal;
  -webkit-box-orient: vertical;
  -webkit-flex-basis: auto;
  -webkit-flex-direction: column;
  -webkit-flex-shrink: 0;
  align-items: stretch;
  border: > 0 solid black;
  box-sizing: border-box;
  display: -webkit-box;
  display: -moz-box;
  display: -ms-flexbox;
  display: -webkit-flex;
  display: flex;
  flex-basis: auto;
  flex-direction: column;
  flex-shrink: 0;
  margin-bottom: 0px;
  margin-left: 0px;
  margin-right: 0px;
  margin-top: 0px;
  min-height: 0px;
  min-width: 0px;
  padding-bottom: 0px;
  padding-left: 0px;
  padding-right: 0px;
  padding-top: 0px;
  position: relative;
  z-index: 0;
}
```

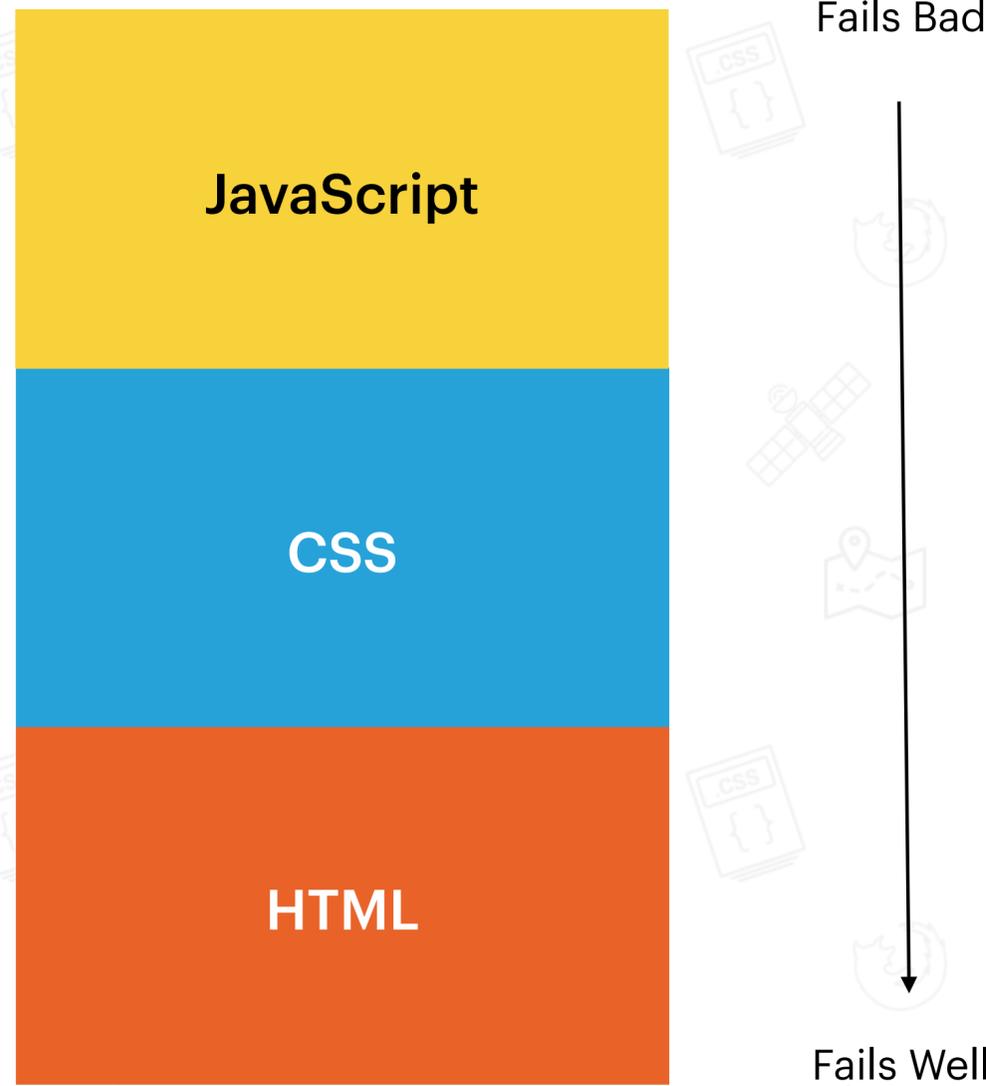
Evaluating Technology

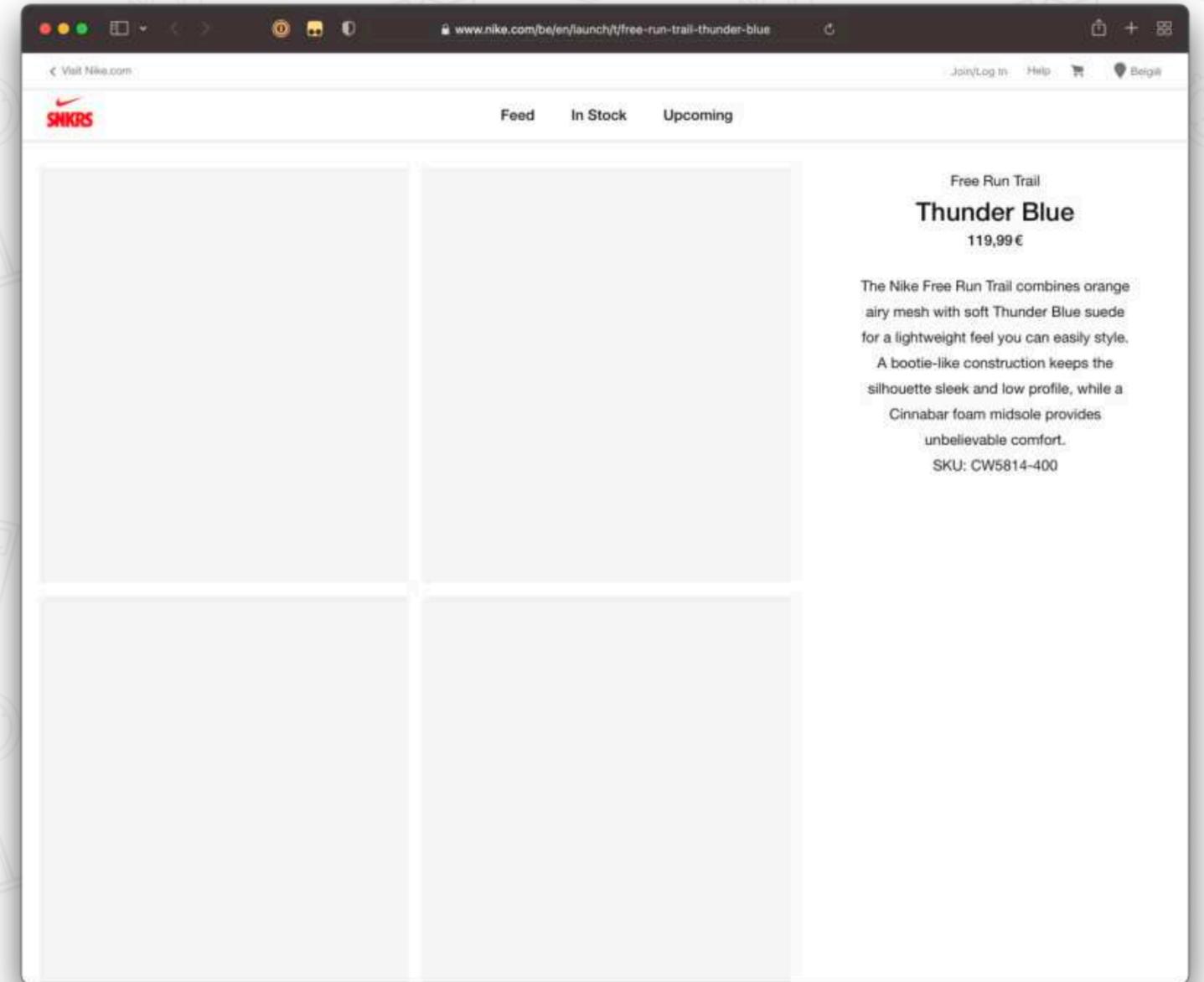
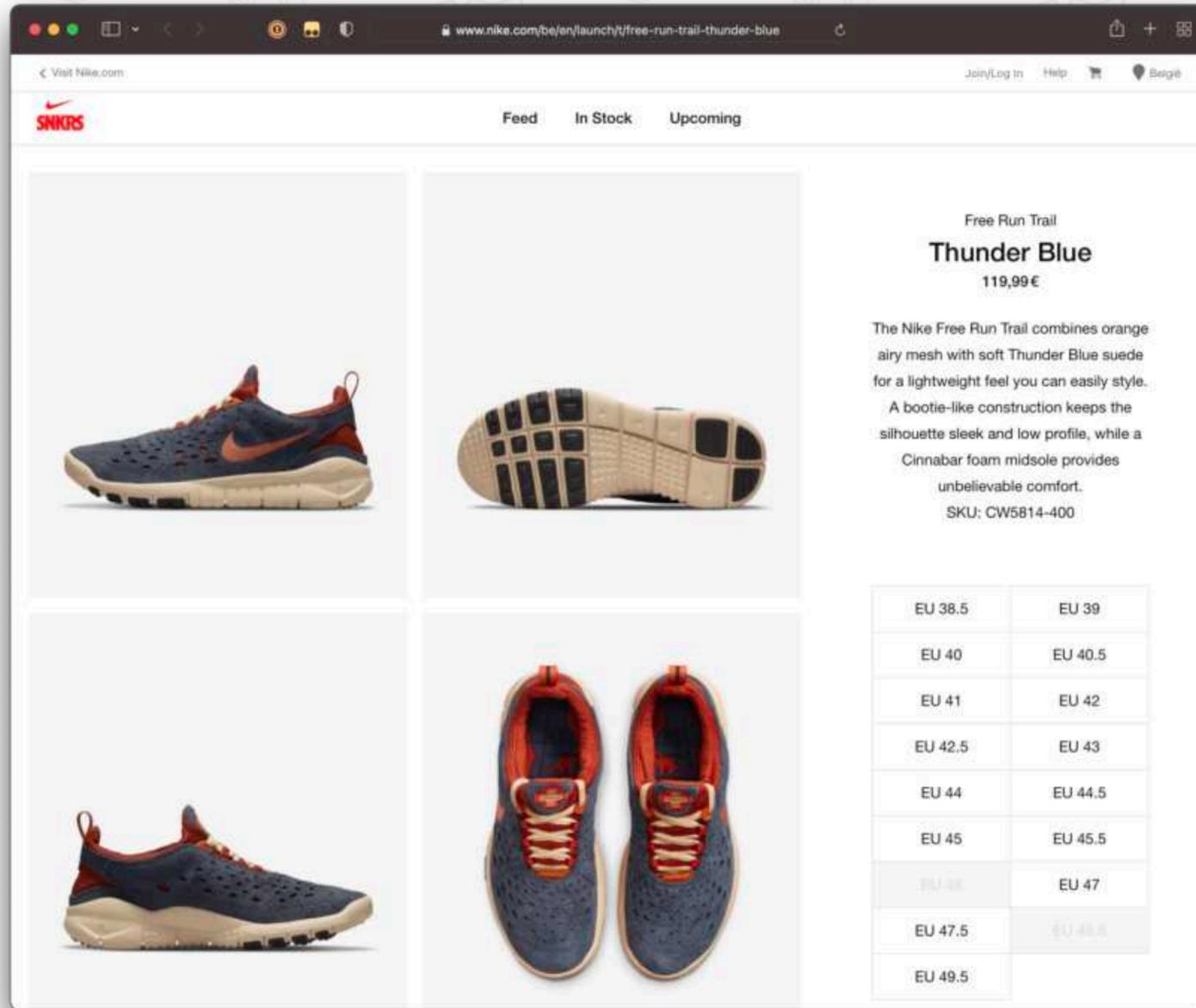
 Jeremy Keith

**We tend to ask 'How well does it work?',
but what you should really be asking is
'How well does it fail?'**

<https://adactio.com/articles/12839>

The Rule of Least Power, on the Web





<https://www.nike.com/be/launch/t/free-run-trail-thunder-blue>

Over-Reliance on JS

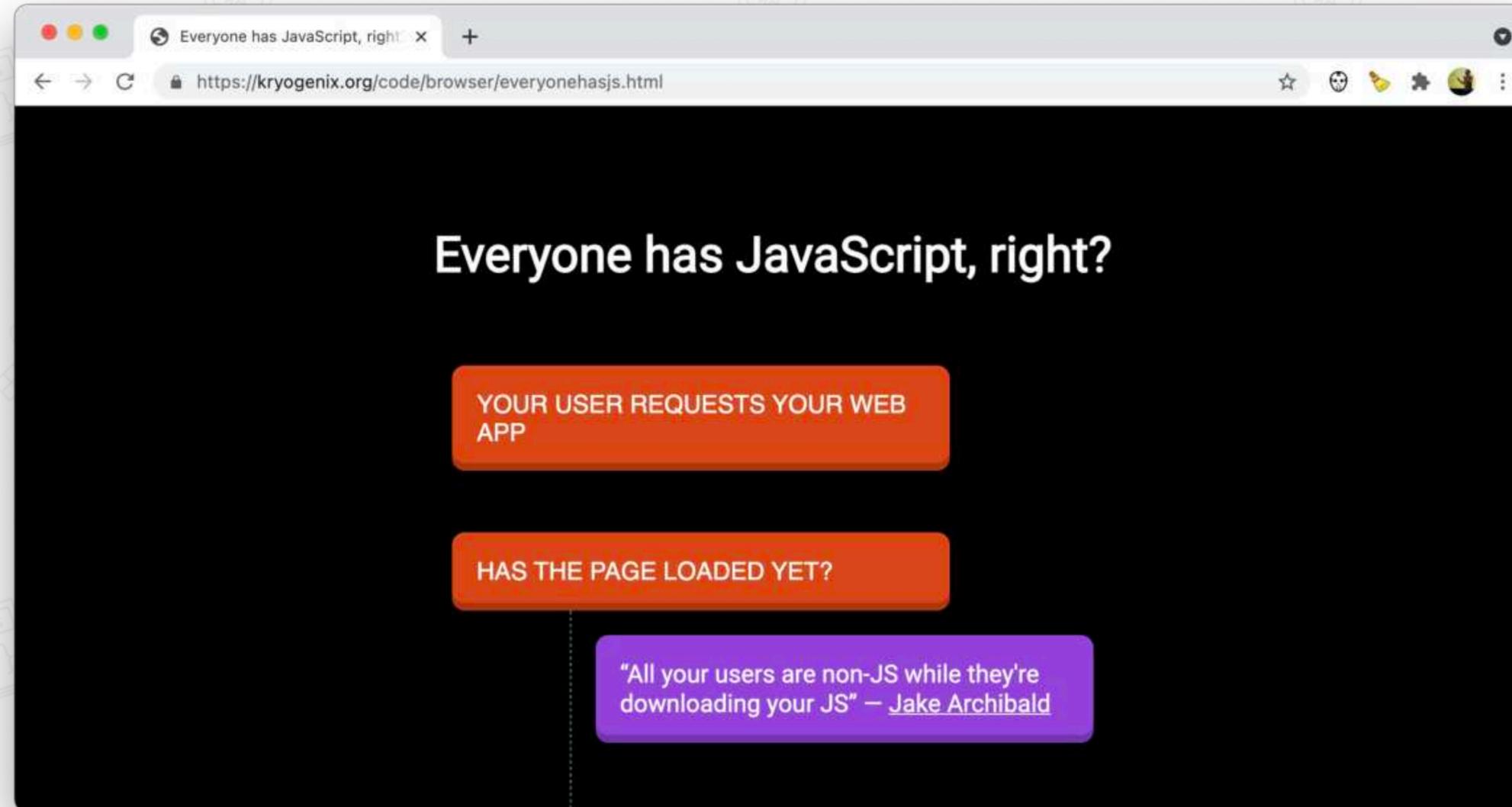
 **Stuart Langridge**

The person not getting all the JavaScript you're serving to them is not someone on some incredibly ancient not-really-a-smartphone in another country with hardly any connection or anything like that. It's you, and it's your customers, in a cellar bar or on a train or while they're waiting for the network to come back up. They **don't have JavaScript turned off.**

<https://kryogenix.org/code/dont-need-that-js/>

Over-Reliance on JS

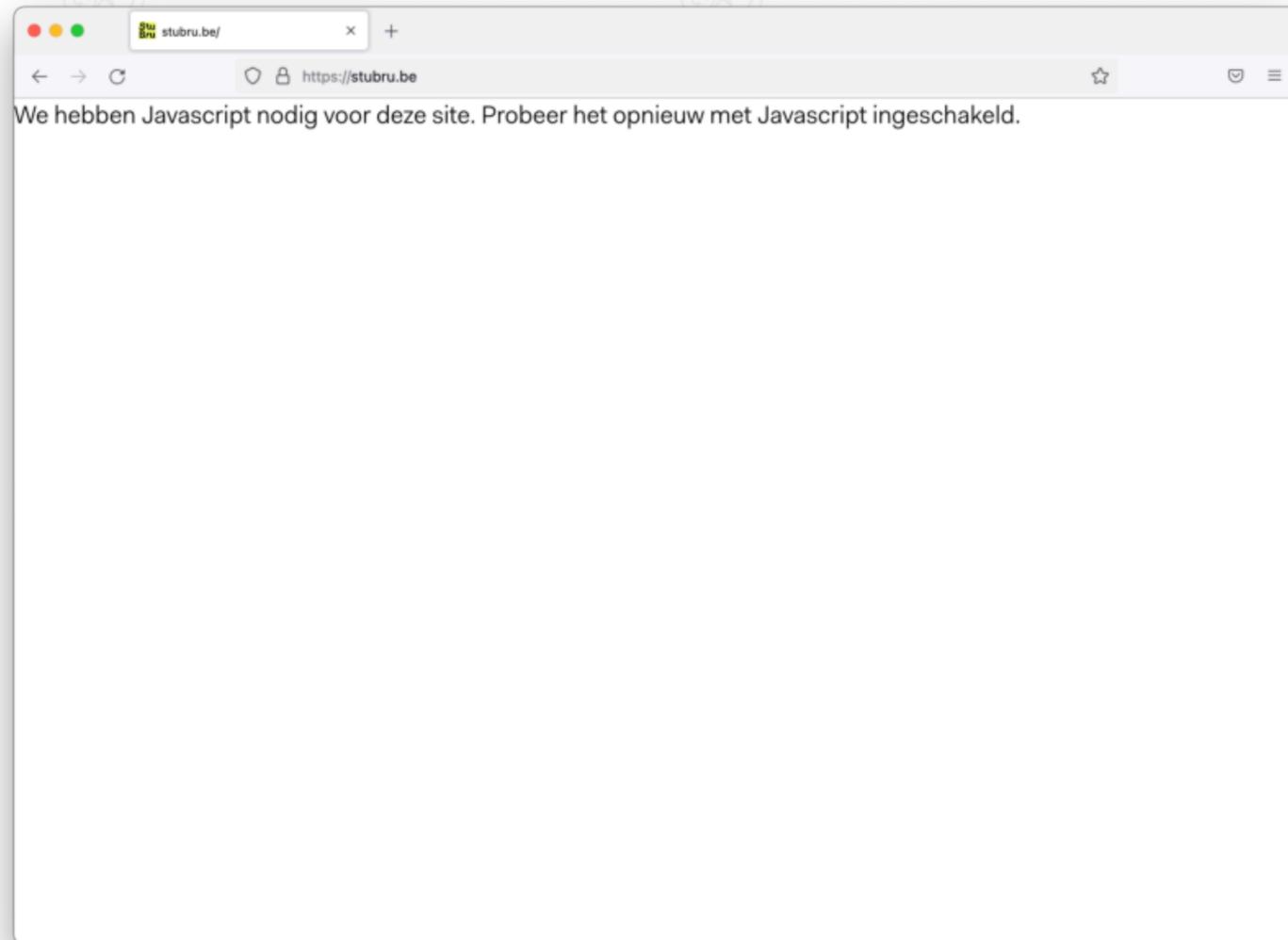
🔗 Everyone has JavaScript, right?



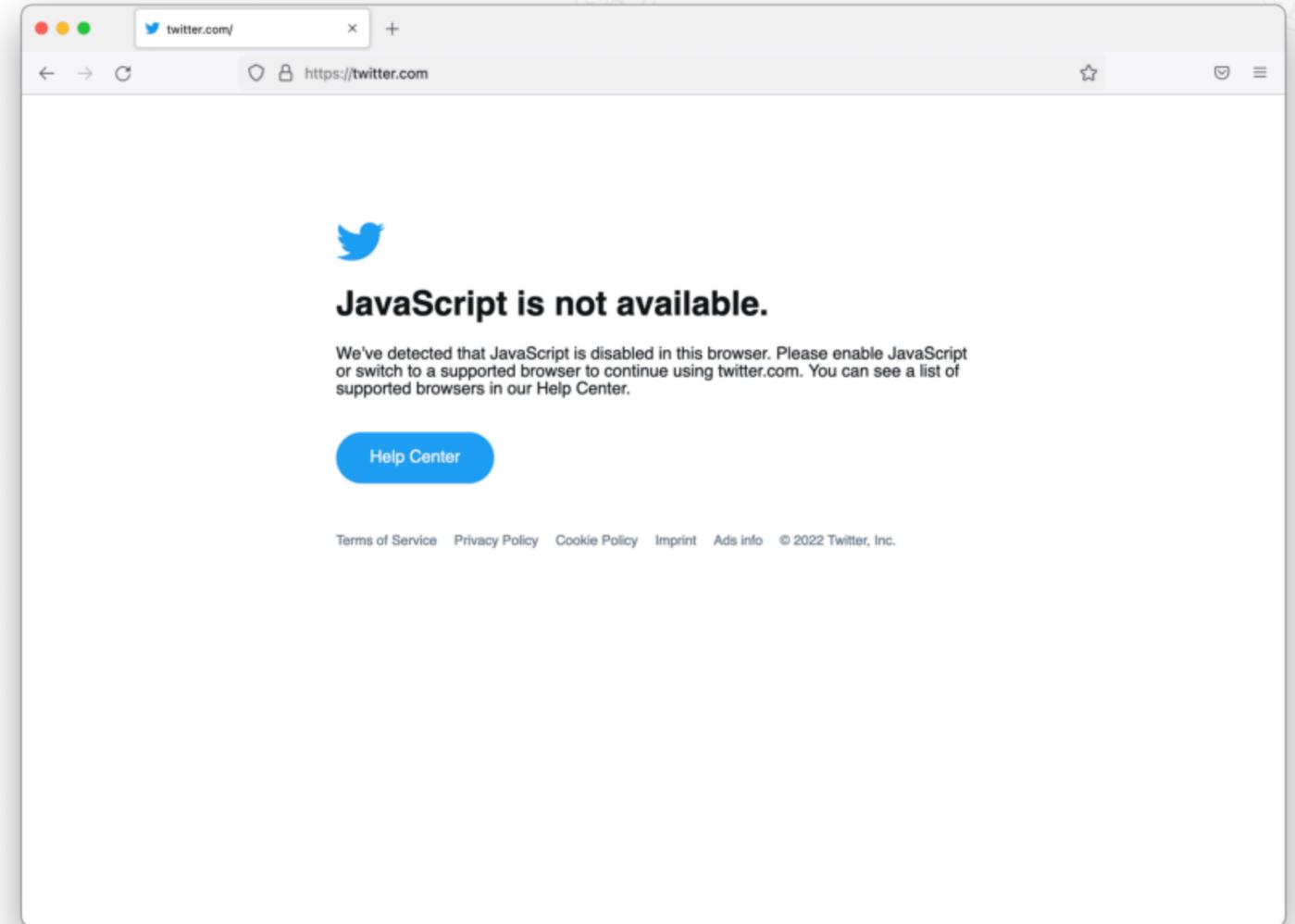
<https://kryogenix.org/code/browser/everyonehasjs.html>

Over-Reliance on JS

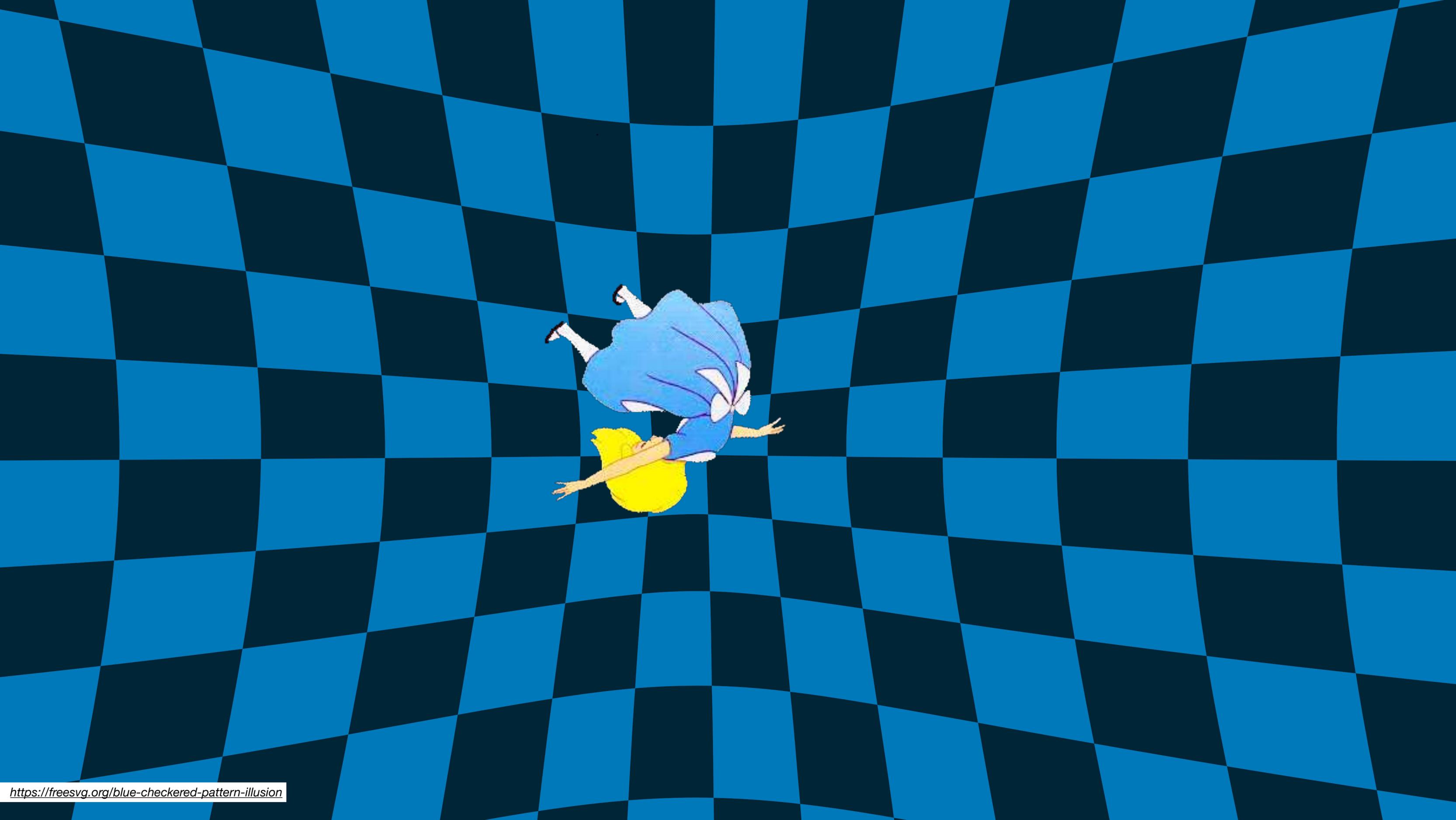
 Even worse ...



<https://www.stubru.be/>



<https://www.twitter.com/>



JavaScript as an Enhancement

Evaluating Technology

 Jeremy Keith

**We tend to ask 'How well does it work?',
but what you should really be asking is
'How well does it fail?'**

<https://adactio.com/articles/12839>

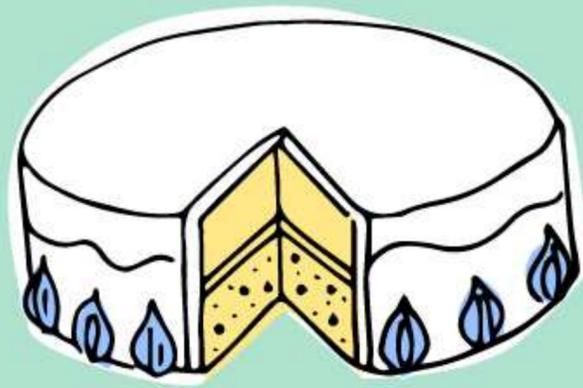
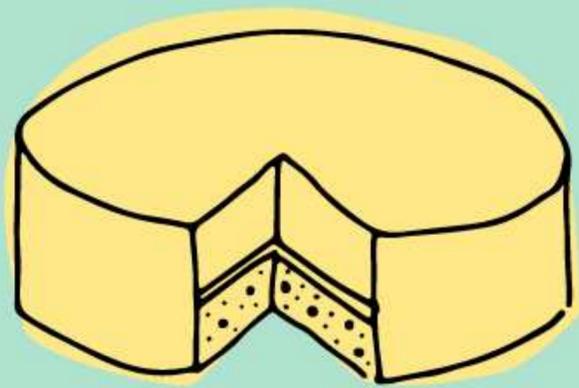
Elevators vs. Escalators

 Jake Archibald

When an elevator fails, it's useless.
When an escalator fails, it becomes stairs.
We should be building escalators, not elevators.

<https://jakearchibald.com/2013/progressive-enhancement-still-important/>

Progressive Enhancement



```
import * as React from 'react';
import PropTypes from 'prop-types';
import Tabs from '@mui/material/Tabs';
import Tab from '@mui/material/Tab';
import Typography from '@mui/material/Typography';
import Box from '@mui/material/Box';

function TabPanel(props) {
  const { children, value, index, ...other } = props;

  return (
    <div
      role="tabpanel"
      hidden={value !== index}
      id={`simple-tabpanel-${index}`}
      aria-labelledby={`simple-tab-${index}`}
      {...other}
    >
      {value === index && (
        <Box sx={{ p: 3 }}>
          <Typography>{children}</Typography>
        </Box>
      )}
    </div>
  );
}
```

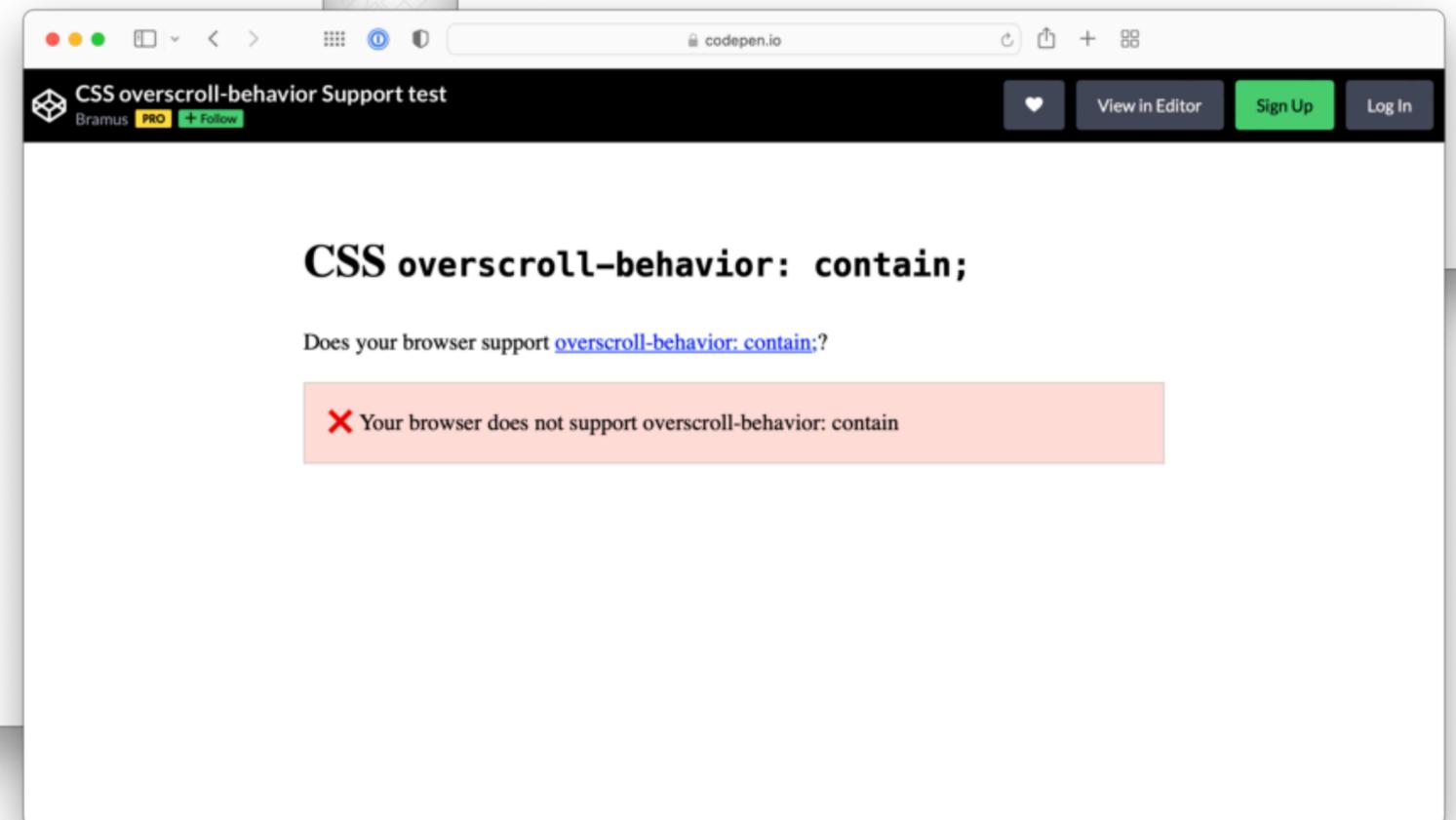
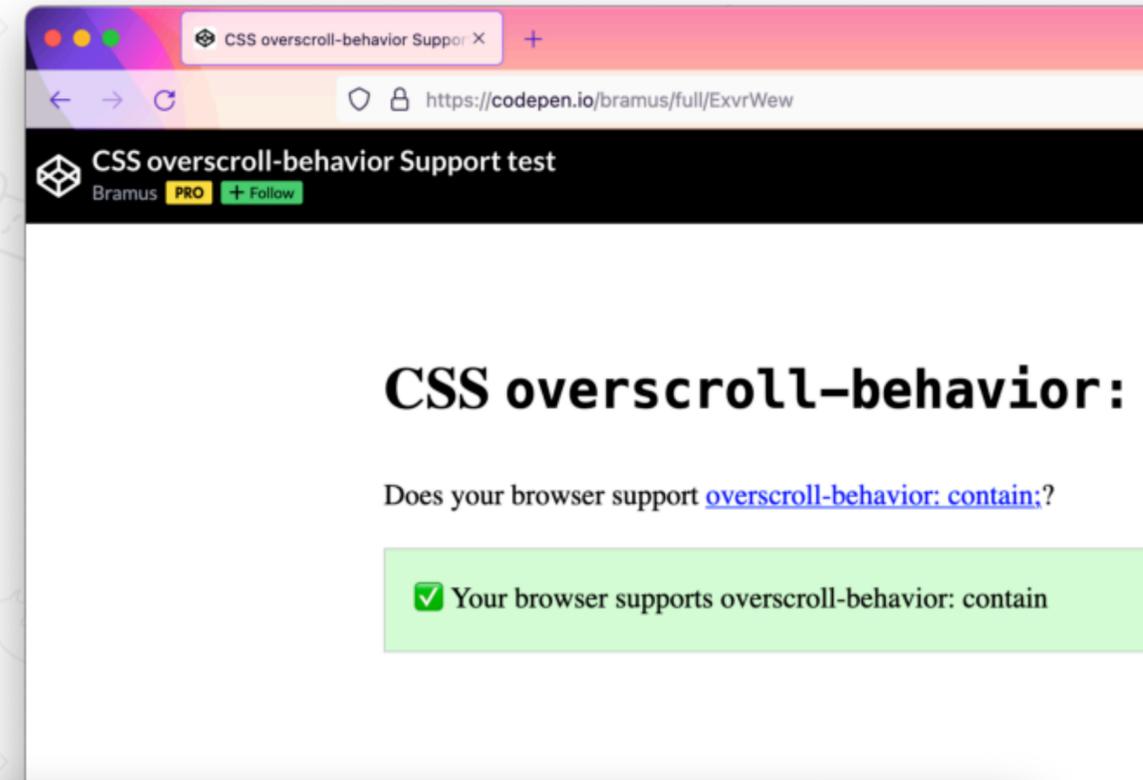
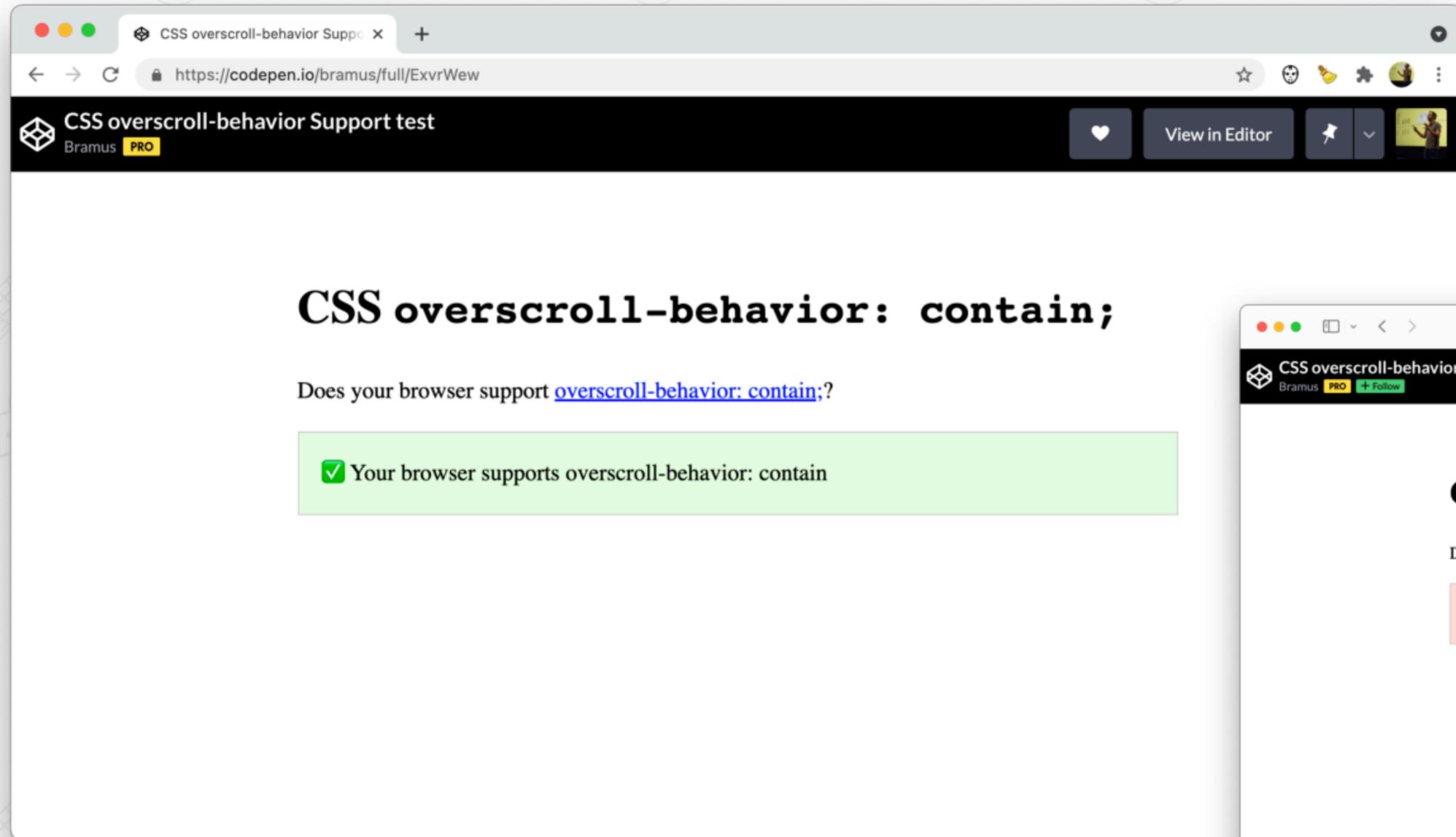
```
✓ [role="tabpanel"] {
  display: none;
}

[name="tabs"]:nth-of-type(1):checked ~ [role="tabpanel"]:nth-of-type(1),
[name="tabs"]:nth-of-type(2):checked ~ [role="tabpanel"]:nth-of-type(2),
[name="tabs"]:nth-of-type(3):checked ~ [role="tabpanel"]:nth-of-type(3) {
  display: block;
}
```

<https://dfkaye.com/posts/2020/08/23/accessible-css-driven-tabs-without-javascript/>

Progressive Enhancement

Feature Detection in CSS with @supports



```
#output::after {
  content: "✘ Your browser does not support overscroll-behavior: contain";
  background-color: #ff00002b;
}

@supports(overscroll-behavior: contain) {
  #output::after {
    content: "✔ Your browser supports overscroll-behavior: contain";
    background-color: #00ff002b;
  }
}
```

Progressive Enhancement

🔗 Feature Detection in JavaScript

- If member in object
- Property on Element
- Method on element return value
- Property on Element retains value

```
if ('geolocation' in navigator) {  
    // Support  
} else {  
    // No Support  
}
```

```
if (!!document.createElement('canvas').getContext) {  
    // Support for canvas  
} else {  
    // No Support  
}
```

Polyfills

 Remy Sharp

A polyfill is a piece of code (or plugin) that provides the technology that you, the developer, expect the browser to provide natively

<https://remysharp.com/2010/10/08/what-is-a-polyfill/>

container-query-polyfill - npm x +

https://www.npmjs.com/package/container-query-polyfill

Nose Picker's Maven

Products Pricing Documentation Community

npm Search packages Search

container-query-polyfill

0.0.9 • Public • Published 20 hours ago

Readme Explore BETA 1 Dependency 0 Dependents 8 Versions

Container Query Polyfill

A tiny polyfill for **CSS Container Queries**, weighing about 1.6kB brotli'd. It transpiles CSS code on the client-side and implements Container Query functionality using **ResizeObserver** and **MutationObserver**.

Usage

Ideally, the polyfill is only loaded if the browser doesn't support Container Queries natively. In a modern setup with a bundler that uses ES modules, the following snippet should work:

```
const supportsContainerQueries = "container" in document.documentElement.style
if (!supportsContainerQueries) {
  import("container-query-polyfill");
}
```

If you are in a legacy setup (or just want to prototype quickly), there's also an IIFE version that you can include using a `<script>` tag:

Install

```
> npm i container-query-polyfill
```

Weekly Downloads

541

Version	License
0.0.9	Apache-2.0

Unpacked Size	Total Files
59.9 kB	9

Last publish

20 hours ago

Collaborators



The Web Catches Up





The Web Catches Up

Examples

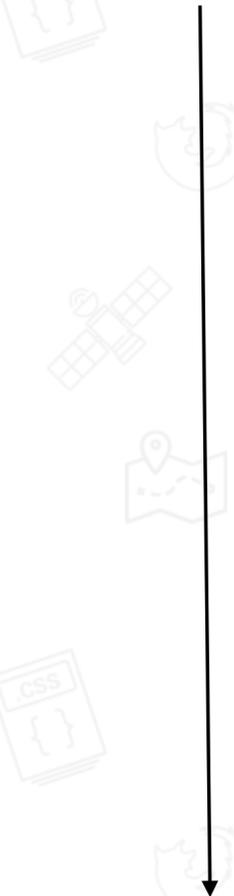
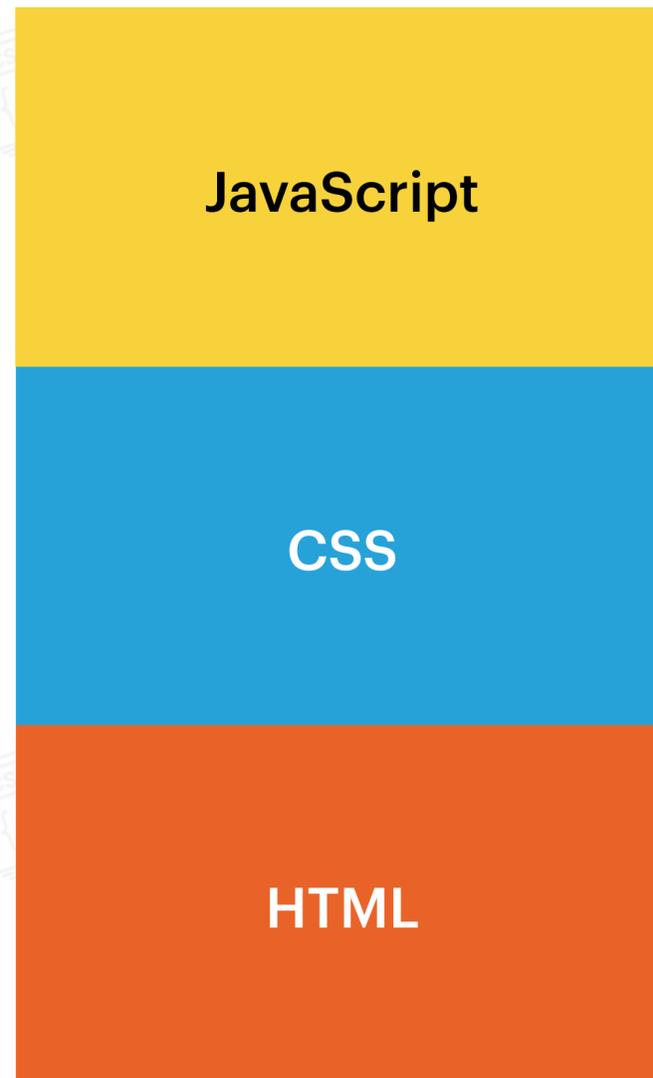
- **Struggling with the very bad `Date()` API in JavaScript?**
👉 The Temporal API is nicer to work with.
- **Using a third-party JavaScript library to animate elements on screen?**
👉 Why not give the built-in Web Animations API a try? It's really powerful.
- **Need colored radio buttons and checkboxes?**
👉 The new CSS `accent-color` property does that for you.

The Web Catches Up

Examples

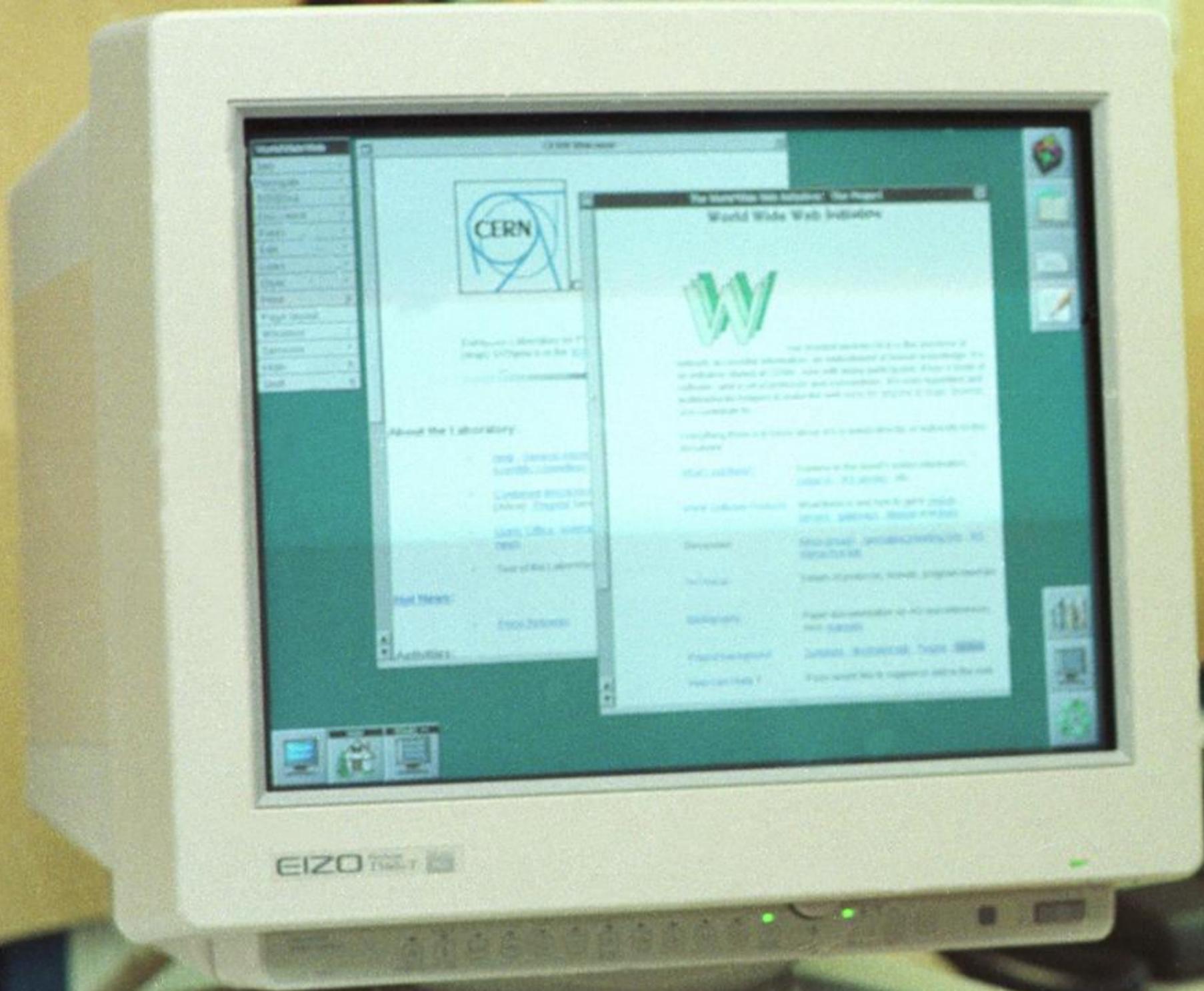
- **Relying on a preprocessor so that you can use CSS variables?**
👉 CSS Custom Properties are the single greatest addition to CSS over the past 20 years.
- **Using Axios?**
👉 Use native fetch, or a wrapper such as Redaxios if you want an Axios-compatible API
- **Scroll-Linked Animations?**
👉 Soon, a problem to be solved using only CSS.

The Web Catches Up



Closer to the core

In Closing



Make better websites

Embrace the Web Platform, don't fight it



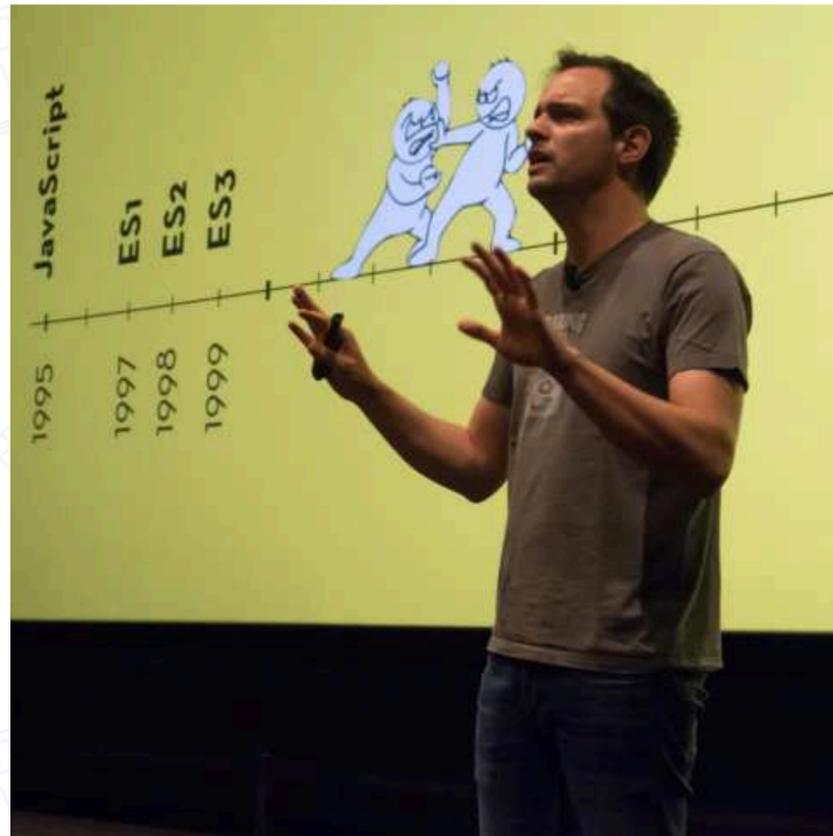
Keep it Simple

Apply the Rule of Least Power

Enhance Progressively

HTML. CSS. JavaScript.
In that order.

Thanks / Questions



[@bramus](https://www.bram.us/)



<https://www.bram.us/>
[@bramusblog](https://www.bram.us/@bramusblog)