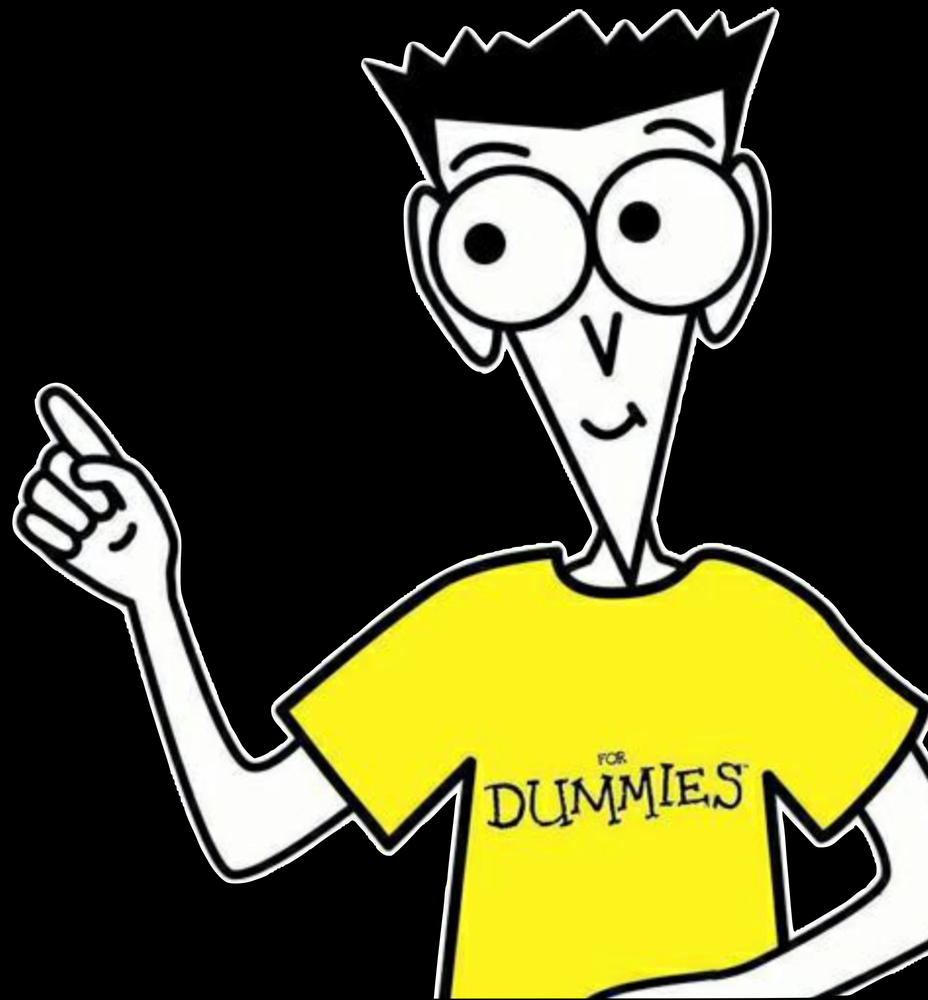
Chris Coyier *presents:*

How to

CSS

1. Write some scope
2. Write some styles



Every selector is scope.

Cool — you've got some styles.

Where and when do you want to apply them?

cssday.nl

Open in Standard ⌘O

Home Schedule Live stream Attendees Workshop Venue

Scope

```
.date {
    color: #222;
    font-size: x-large;
    margin: 0 .4em;
}

@media only screen and (min-width: 48em) {
    .date {
        margin: 0;
    }
}
```

Scope

Scope

The Real World

CODEPEN

Search CodePen...

Sign Up

Log In

TRY OUR ONLINE EDITOR

Start Coding

Search Pens

Challenges

Spark

CodePen

The best place to build, test, and discover front-end code.

CodePen is a **social development environment** for front-end designers and developers. Build and deploy a website, show off your work, build test cases to learn and debug, and find inspiration.

Sign Up for Free

SPONSOR

MongoDB.

MongoDB Atlas

Start building for free with MongoDB Atlas—the fast and secure multi-cloud database.

HTML

```
<div class="rect"></div>
```

SCSS

```
.rect {
    background: linear-gradient(
        -119deg,
        $gray 0%,
        $dark-gray 100%); }
```

JS

```
var colors =
["#74B087","#DE7300","#74B087"];

function animate() {};
```

Build & Test

Get work done quicker by building out **entire projects** or isolating code to **test features and animations**. Want to keep it all under wraps? **Upgrade to a PRO account** to keep your work private.

Try the Editor

Learn & Discover

Want to upgrade your skills and get noticed? Participating in CodePen Challenges is a great way to try something new. We frequently feature these Pens on our homepage and across social media!

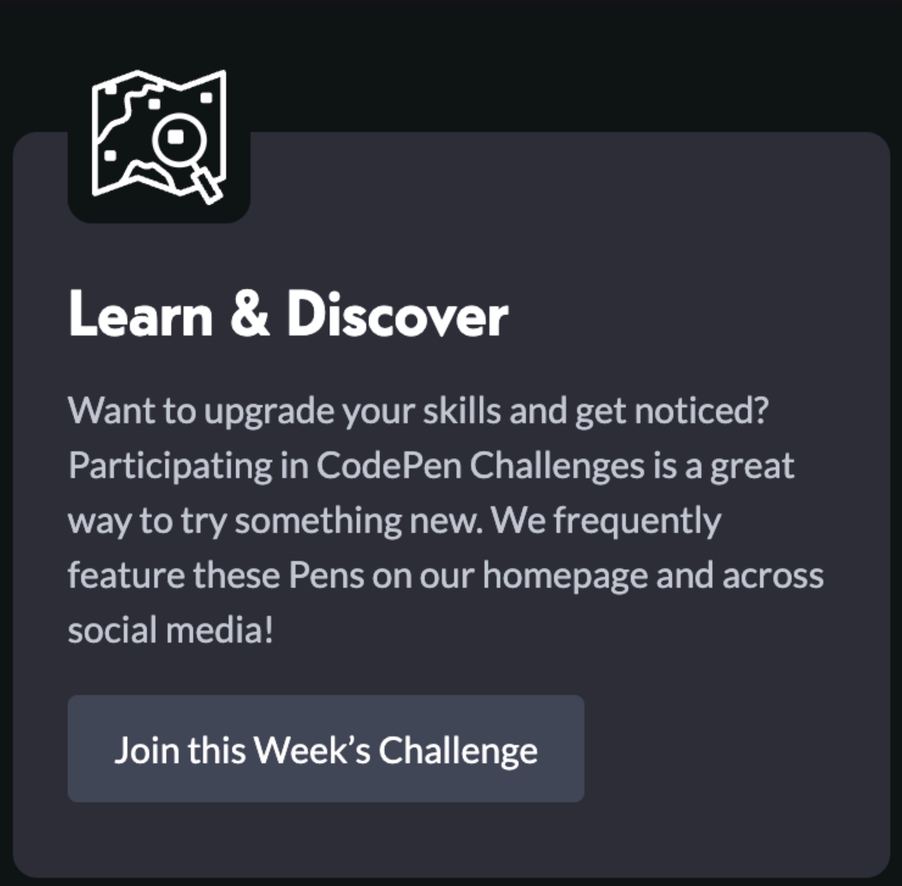
Join this Week's Challenge

Share Your Work

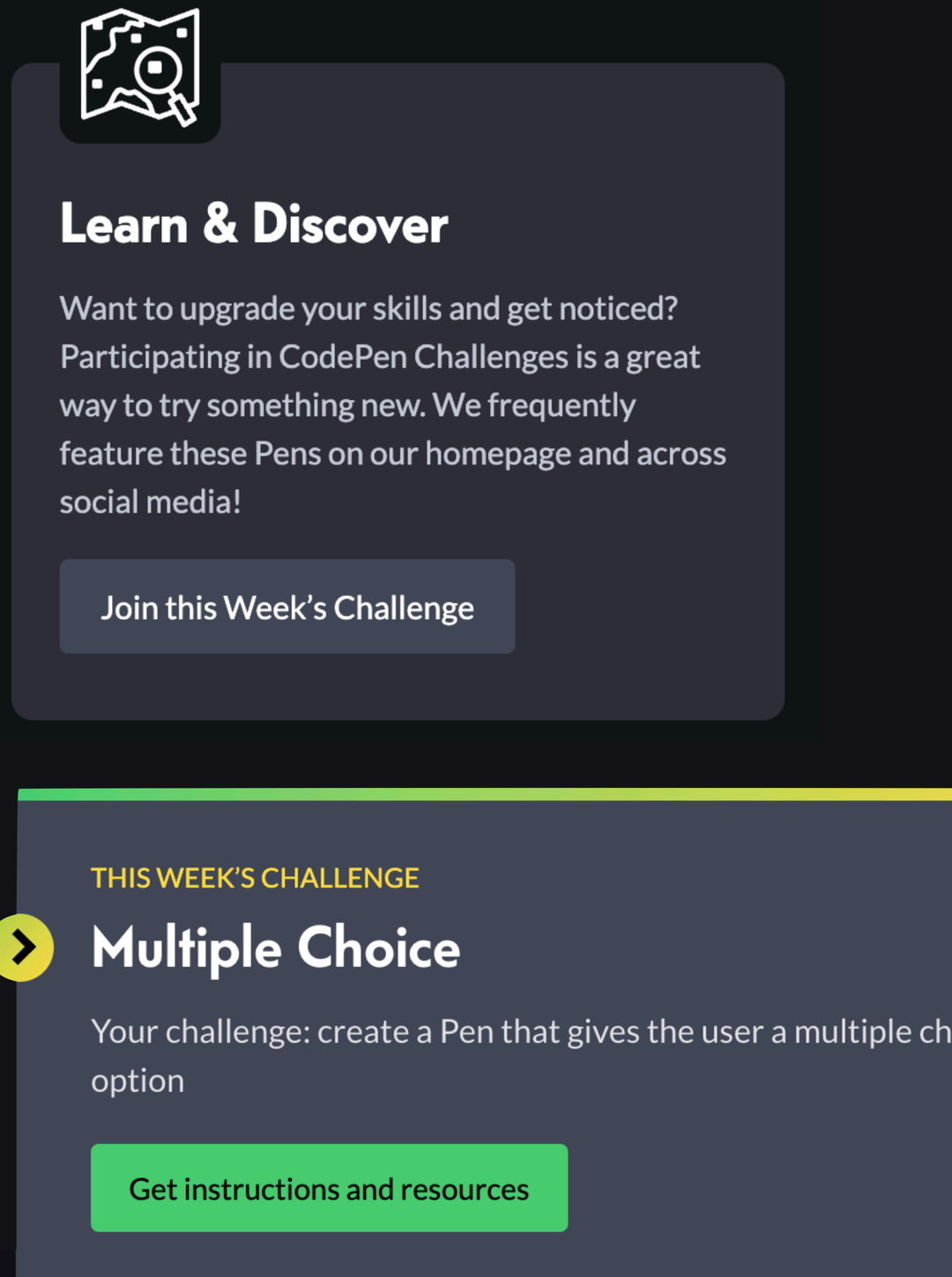
Become a part of the **most active front-end community** in the world by sharing work. Presenting at a conference? Show your code directly in the browser with **Presentation Mode**.

Explore Trending

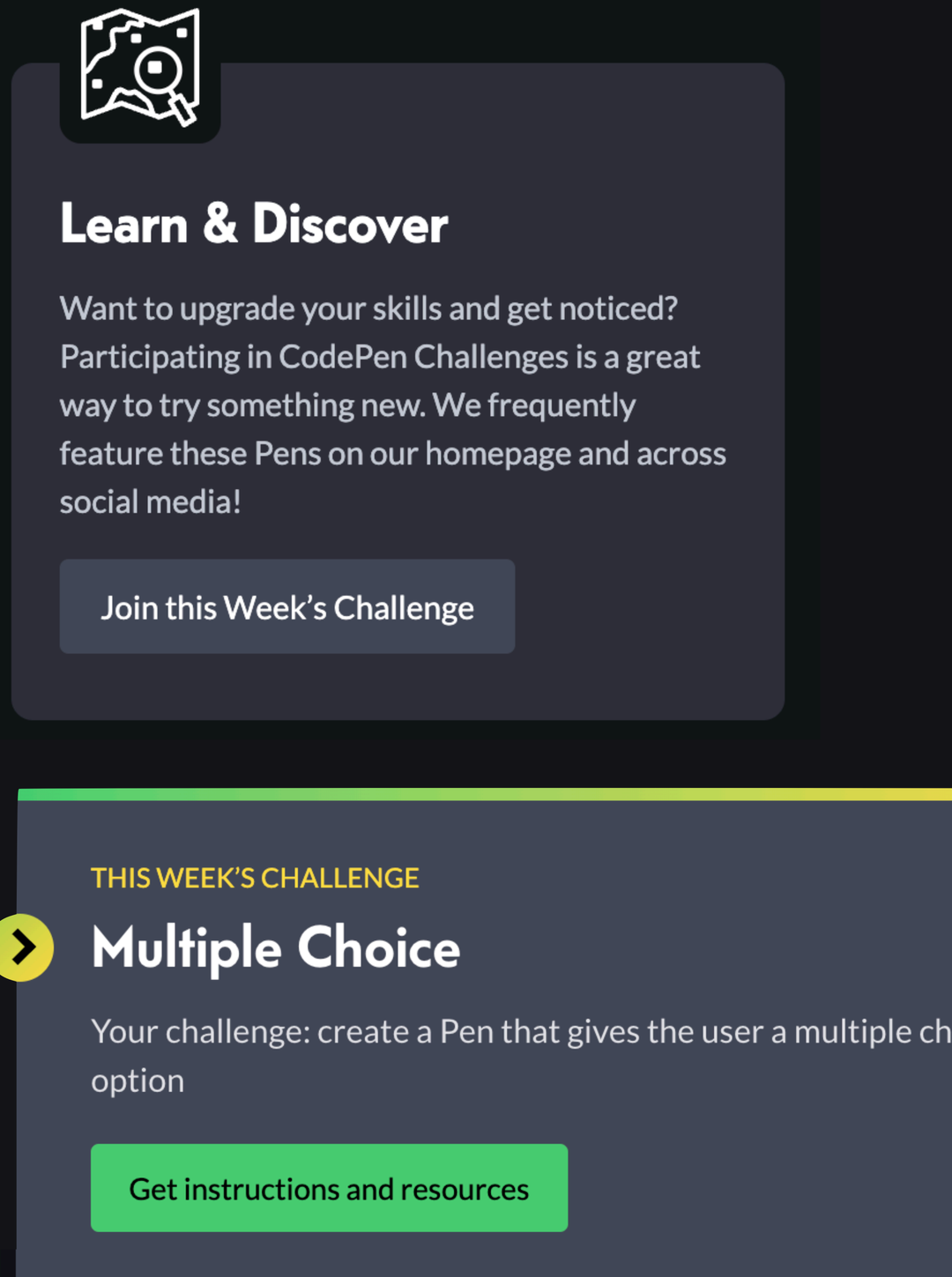
```
.card {
  background: gray;
  color: #eee;
  border-radius: 10px;
  padding: 1rem;
}
```



```
.card {
  background: gray;
  color: #eee;
  border-radius: 10px;
  padding: 1rem;
}
```



```
.challenge-card {
  background: lightgray;
  color: #eee;
  padding: 1.5rem;
}
```



If that's all the scoping you ever do in CSS, that's... fine.

Really, actually, **very fine.** 👍

The only reason to move on to more advanced scoping is when you're **actually experiencing problems.**

One problem could be that you forgot that you used `card` as a class already, redeclared it for a new use, and got weird results.

Or that you work on a team and someone else does that and is confused, doesn't realize it, or messes with the other CSS out of that **confusion**.

```
.card {
  ...
}
```

Learn & Discover

Want to upgrade your skills and get noticed? Participating in CodePen Challenges is a great way to try something new. We frequently feature these Pens on our homepage and across social media!

Join this Week's Challenge

```
.card {
  ...
}
```

THIS WEEK'S CHALLENGE

Multiple Choice

Your challenge: create a Pen that gives the user a multiple ch
option

Get instructions and resources

🧠 **CSS BRAIN** 😵‍💫

`text-wrap: pretty;`

Writing CSS can have unintended consequences.

There are tools to take the idea of scoping further.

be like water yo.

— John Allsop

scope the snot out of everything.

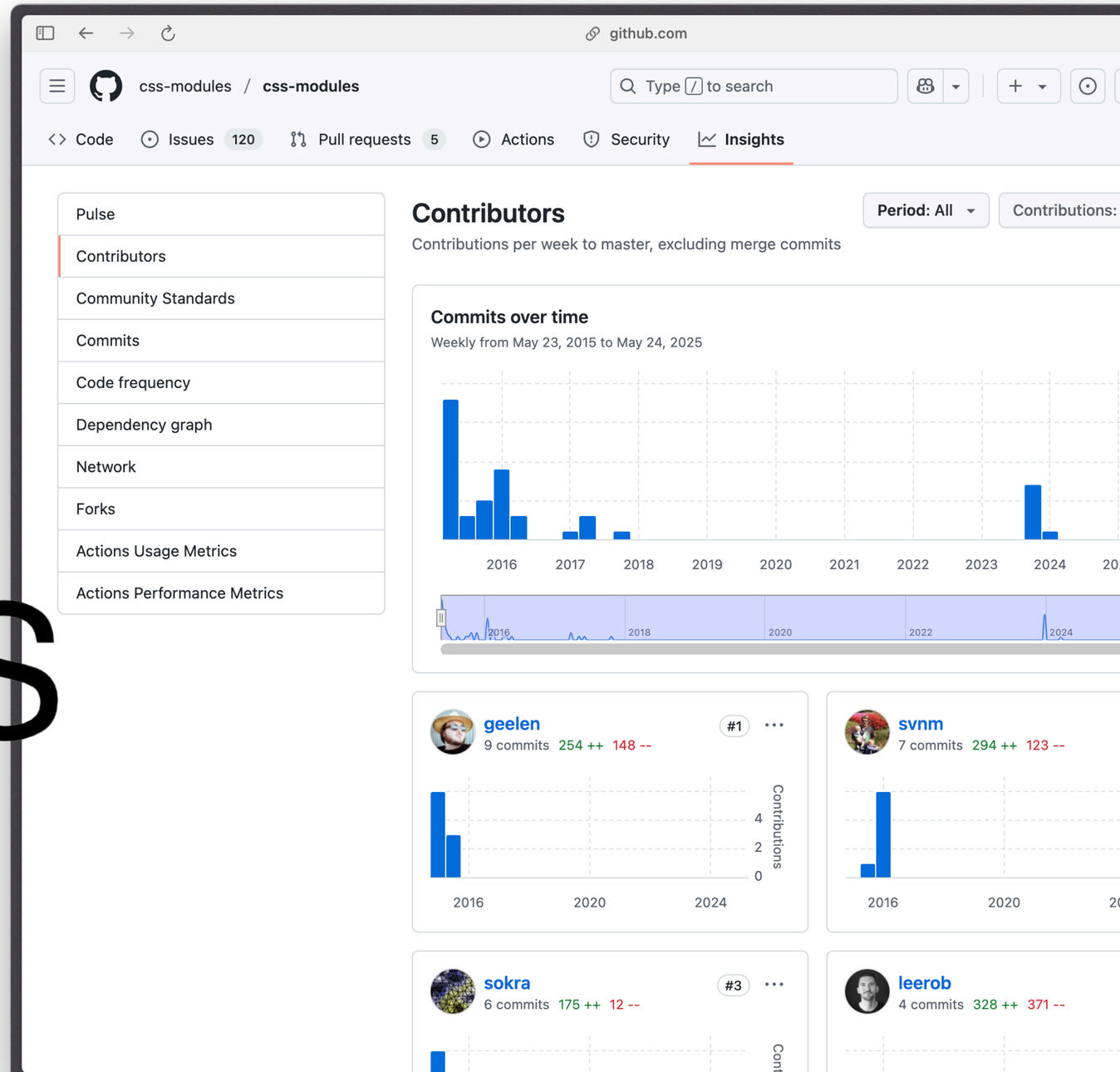
— me in this talk

There is web platform stuff. We have `@scope` now in CSS and I promise we'll talk about that — later.

There are tools that *process* HTML/CSS/JavaScript in order to help with scoping. Some of them are explicitly about selector scoping and for some it's an implied benefit.

CSS MODULES

🎂 10 year birthday!



CSS modules *implies a build process* for the CSS and is exclusively for JavaScript-produced HTML.

card.module.css

```
.card {
  background: gray;
  color: #eee;
  border-radius: 10px;
  padding: 1rem;
}
```

Produced CSS

```
.card-Adf98xd {
  background: gray;
  color: #eee;
  border-radius: 10px;
  padding: 1rem;
}
```

What is imported

```
export default const styles = {
  "card": "card-Adf98xd"
}
```

Component.js

```
import styles from "./card.module.css";

element.innerHTML = `
  <div class="' + styles.card + '">
`;
```

Produced HTML

```
<div class="card-Adf98xd">
```

Produced HTML

```html
<div class="card-Adf98xd">
```

Produced CSS

```css
.card-Adf98xd {
  background: gray;
  color: #eee;
  border-radius: 10px;
  padding: 1rem;
}
```

CSS Modules is just a spec. You won't see it in your `package.json` file. Other tech just builds it in.

Next.js	Nuxt.js	Bun
Astro	webpack css-loader	Lightning CSS
Vue SFCs	Parcel	Rspack
Gatsby	Vite	PostCSS
SvelteKit	create-react-app	Solid

ModalFileUploadErrors.tsx — cp

ModalFileUploadErrors.tsx — cp

ModalFileUploadErrors.tsx 9+

cpeditor > components > ModalFileUploadErrors > ModalFileUploadErrors.tsx > ModalFileUploadErrors

You, 3 weeks ago | 2 authors (You and one other)

```
1  import {
2    Button,
3    BUTTON_COLORS,
4    MODAL_ALERT_TYPES,
5    ModalAlert,
6    Table
7  } from '@codepen/library';
8  import { humanReadableFilesize, isString } from '@codepen/util';
9
10 import styles from './ModalFileUploadErrors.module.scss';
11
12 export default function ModalFileUploadErrors(props: {
13   closeModal: () => void;
14 }) {
15
16   return (
17     <ModalAlert
18       {...props}
19       type={MODAL_ALERT_TYPES.DANGER}
20       title="File Upload Errors"
21       className={styles.root}
22       footer={
23         <>
24           <Button
```

Import

Use

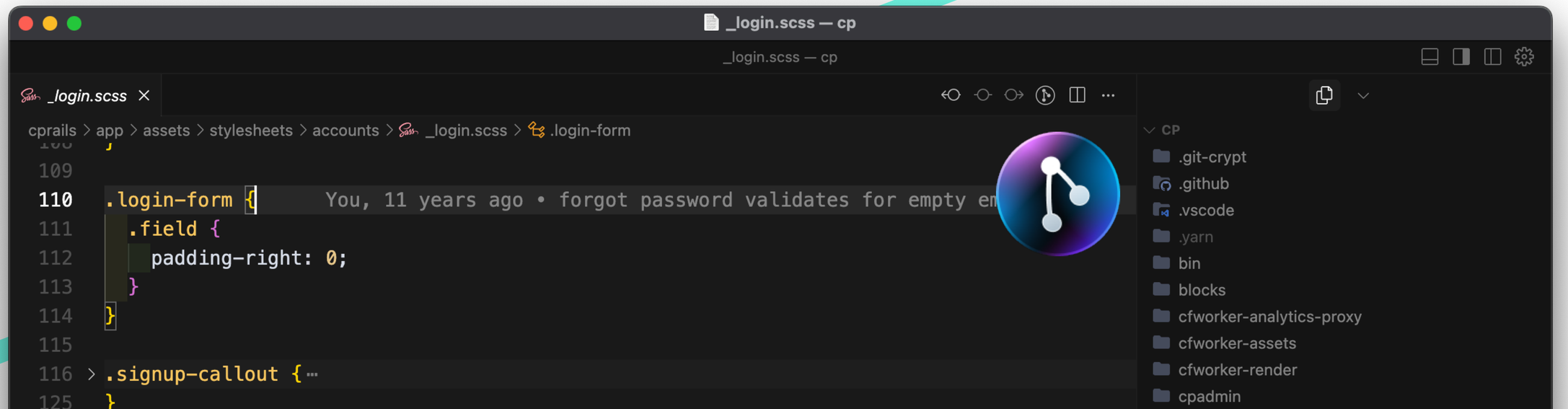
CP

- cpeditor
  - components
    - HomepageShared
    - HomepageSharedInviteCount
    - HomepageSharedInvites
    - InviteCollaborator
    - KeyList
    - LivePreviewButton
    - ModalAddToCollection
    - ModalBlocks
    - ModalColumns
    - ModalConfirmCollaboratorChange
    - ModalConfirmDeleteFileEntry
    - ModalConfirmRemoveDomain
    - ModalConfirmRemovePenBlock
    - ModalConfirmUndeployPen
    - ModalFileUploadDuplicates
    - ModalFileUploadErrors
      - index.js
      - ModalFileUploadErrors.module....
      - ModalFileUploadErrors.tsx 9+
    - ModalKeyboardShortcuts
    - ModalKeyboardShortcutsOmniba...
    - ModalLoginSignup
    - ModalPromptFork

Co-locate the component styles

Orthogonal Problem

Scoping CSS is a related concept of **unused CSS**. Performance tools can bitch at you about unused CSS but it's hard to accurately detect site-wide and teams can be scared to touch global CSS because of unknown consequences.



Unused CSS is less of a concern with scoped and co-located styles.

Toss the component — or don’t use it — the styles aren’t loaded.

CSS-in-JS is scoping, too.

Styled Components

Emotion

JSS

styled-jsx

Vanilla Extract

Stitches

Linaria

CSS Modules

Styletron

Fela

Goober

StyleX

Radium

Aphrodite

Global API
Application
General

Composition API
setup()
Reactivity: Core
Reactivity: Utilities
Reactivity: Advanced
Lifecycle Hooks
Dependency Injection
Helpers

Options API
Options: State
Options: Rendering
Options: Lifecycle
Options: Composition
Options: Misc
Component Instance

Built-ins
Directives
Components
Special Elements
Special Attributes

Single-File Component
Syntax Specification
<script setup>
CSS Features

Advanced APIs
Custom Elements
Render Function
Server-Side Rendering

SFC CSS Features

Scoped CSS

When a `<style>` tag has the `scoped` attribute, its CSS will apply to elements of the current component only. This is similar to the style encapsulation found in Shadow DOM. It comes with some caveats, but doesn't require any polyfills. It is achieved by using PostCSS to transform the following:

```vue
<style scoped>
.example {
  color: red;
}
</style>

<template>
  <div class="example">hi</div>
</template>
```

Into the following:

```vue
<style>
.example[data-v-f3f3eg9] {
  color: red;
}
</style>

<template>
  <div class="example" data-v-f3f3eg9>hi</div>
</template>
```

PostCSS to transform the following:

```
<style scoped>
.example {
  color: red;
}
</style>

<template>
  <div class="example">hi</div>
</template>
```

Into the following:

```vue
<style>
.example[data-v-f3f3eg9] {
  color: red;
}
</style>

<template>
  <div class="example" data-v-f3f3eg9>hi</div>
</template>
```

Keep the class *and* scope the styles.

Introduction
Overview
Getting started
.svelte files
.svelte.js and .svelte.ts files

Runes
What are runes?
$state
$derived
$effect
$props
$bindable
$inspect
$host

Template syntax
Basic markup
{#if ...}

SVELTE • STYLING

Scoped styles

ON THIS PAGE

Svelte components can include a `<style>` element containing CSS that belongs to the component. This CSS is *scoped* by default, meaning that styles will not apply to any elements on the page outside the component in question.

This works by adding a class to affected elements, which is based on a hash of the component styles (e.g. `svelte-123xyz`).

```
<style>
  p {
    /* this will only affect <p> elements in this component */
    color: burlywood;
  }
</style>
```

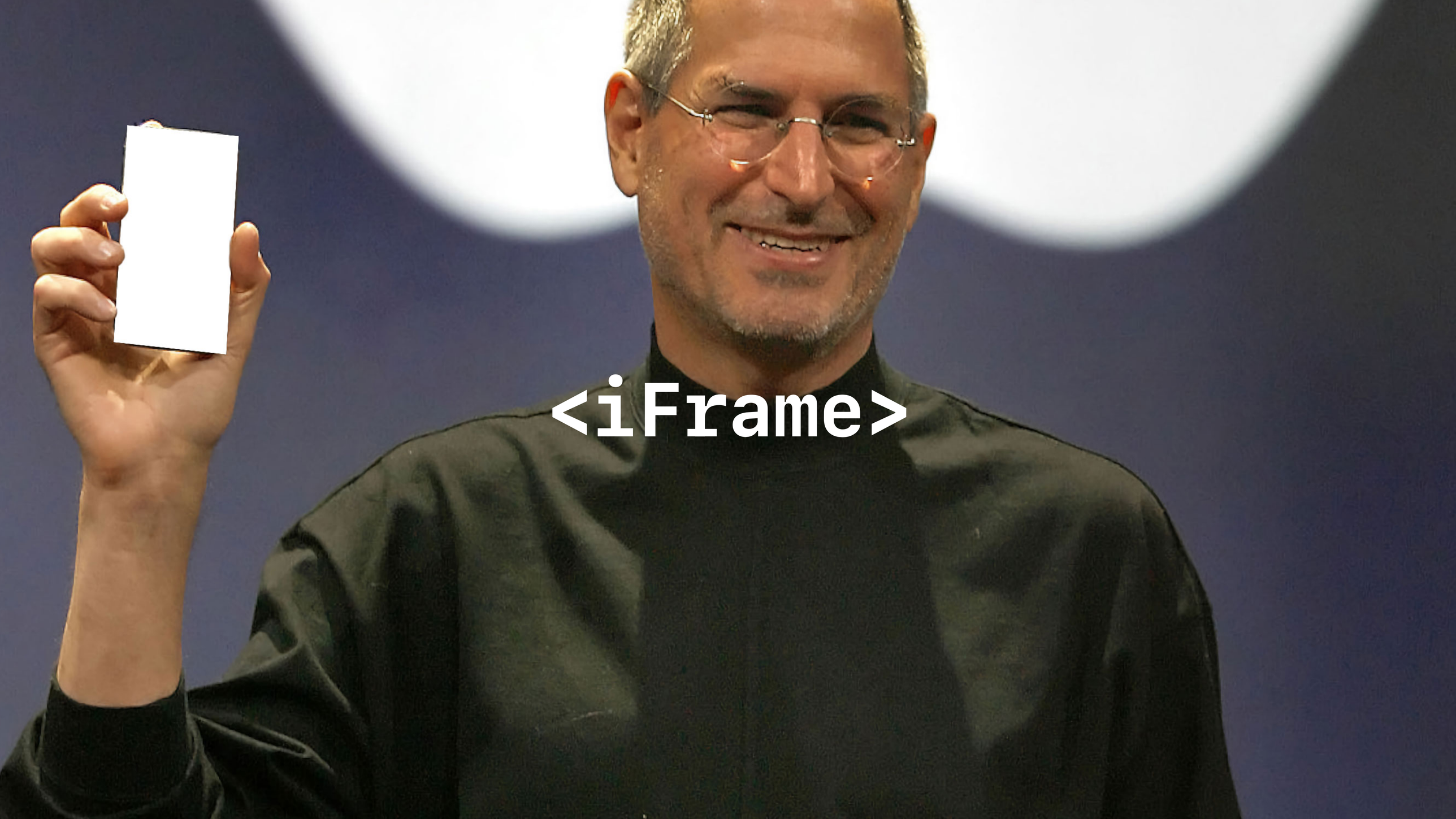
The web platform doesn't really help us with this kind of selector scoping.

It goes from very lightweight to **very extreme**.

Hardcore SCOPING

<iframe>

<iFrame>


<iFrame>

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My New Pen</title>
    <link rel="stylesheet" href="./style.css">
  </head>
  <body>
    <main>
      <iframe src="https://www.cssday.nl/"></iframe>
    </main>

    <script src="./script.js"></script>
  </body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My New Pen</title>
    <link rel="stylesheet" href="./style.css">
  </head>
  <body>
    <main>
      <object data="./content.html"></object>
    </main>

    <script src="./script.js"></script>
  </body>
</html>
```

Shadow DOM

```
<svg style="display:none" xmlns=
  <defs>
    <symbol id="user-icon" viewB
      <path d="M25.333 9.335c0 5
9.335zM23.203 18.908c-2.008 1.51
13.095h32c0-3.231-3.467-10.675-8
    </symbol>
  </defs>
</svg>

<div class="icon">
  <svg width="1em" height="1em">
    <use xlink:href="#user-icon"
  </svg>
</div>
```

HTML

```
13.092z">
    </symbol>
  </defs>
</svg>

<div class="icon">
  <svg width="1em" height="1em">
    <use xlink:href="#user-icon"></use>
  </svg>
</div>
```

CSS

JS

Console Assets Comments ⌘ Last saved LESS THAN A MINUTE AGO Delete Add to Collection

Elements Console Sources Network Performance Memory Application Lighthouse Apollo

```
<!DOCTYPE html>
<html lang="en" class>
  <head> ... </head>
  <body> grid
    <svg style="display:none" xmlns="http://www.w3.org/2000/svg"> ... </svg>
    <div class="icon">
      <svg width="1em" height="1em">
        <use xlink:href="#user-icon">
          #shadow-root (user-agent) == $0
            <svg id="user-icon" viewBox="0 0 32 32">
              <path d="M25.333 9.335c0 5.153-4.179 9.333-9.333 9.333s-9.333-4.18-9.333-
              9.333c0-5.156 4.179-9.335 9.333-9.335s9.333 4.179 9.333 9.335zM23.203 18.9
              08c-2.008 1.516-4.499 2.427-7.203 2.427-2.707 0-5.199-0.913-7.209-2.429-5.
              429 2.391-8.791 9.835-8.791 13.095h32c0-3.231-3.467-10.675-8.797-13.092z">
              </path>
            </svg>
        </use>
```

Styles

Filte

Web Components allow you to make your own Shadow DOM.

They don't *need* to use Shadow DOM, and honestly I often prefer not to, but it is a powerful scoping tool unique to **the platform**.

```
class FancyButton extends HTMLElement {
  connectedCallback() {
    this.attachShadow({ mode: "open" });

    const style = document.createElement("style");
    style.textContent = `
      button {
        background: #1E88E5;
        color: white;
        padding: 2rem 4rem;
        border: 0;
        font-size: 1.5rem;
        margin-block-end: 2rem;
      }
    `;
    this.shadowRoot.appendChild(style);

    this.shadowRoot.appendChild(this.firstElementChild);
  }
}

window.customElements.define("fancy-button", FancyButton);
```

```
<fancy-button>
  <button>Web Component Button</button>
</fancy-button>
```

Web Component Button

```
<button>Not Web Component Button</button>
```

Not Web Component Button

Web Components are pretty sweet for design systems. Theoretically? No, *actually*.

Web Components are pretty sweet for design systems. Theoretically? No, *actually*.

Google Material Design

Adobe Spectrum

Microsoft FAST

IBM Carbon

Salesforce Lightning

Shoelace / Web Awesome

UTILITY CLASS SCOPING

Styling via classes that do one thing each.

without ever leaving your HTML.

text-lg text-white font-medium

A utility-first CSS framework packed with classes like `flex`, `pt-4`, `text-center` and `rotate-90` that can be composed to build any design, directly in your markup.

Get started

Quick search ⌘K

```
<div class="flex flex-col items-center gap-6 p-7 md:flex-row md:gap-8 rounded-2xl">
  <div>
    <img class="size-48 shadow-xl rounded-md" alt="" src="/img/cover.png" />
  </div>
  <div class="flex items-center md:items-start">
    <span class="text-2xl font-medium">Class Warfare</span>
    <span class="font-medium text-sky-500">The Anti-Patterns</span>
    <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
      <span>No. 4</span>
      <span>·</span>
      <span>2025</span>
    </span>
  </div>
</div>
```

Class Warfare

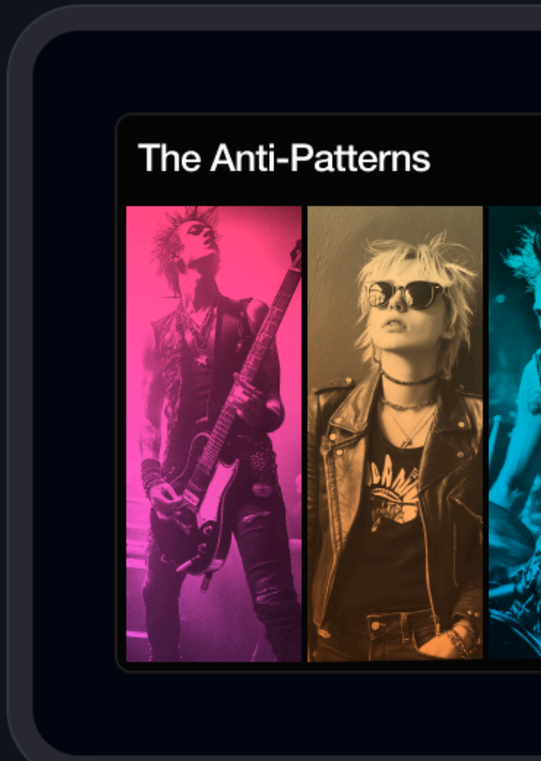
The Anti-Patterns

No. 4 · 2025

With Tailwind, you're mostly not writing CSS. CSS is a byproduct of HTML classes.

Utility styling classes applied directly to HTML is inherently tightly scoped. You can't accidentally style something else.

```html
<div class="flex flex-col items-center gap-6 p-7 md:flex-row md:gap-8 rounded-2xl">
  <div>
    <img class="size-48 shadow-xl rounded-md" alt="" src="/img/cover.png" />
  </div>
  <div class="flex items-center md:items-start">
    <span class="text-2xl font-medium">Class Warfare</span>
    <span class="font-medium text-sky-500">The Anti-Patterns</span>
    <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
      <span>No. 4</span>
      <span>·</span>
      <span>2025</span>
    </span>
  </div>
</div>
```



ok ok ok

Vanilla CSS

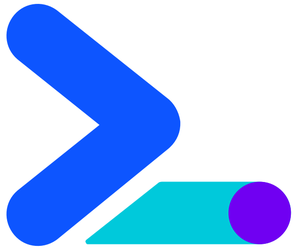
@scoop

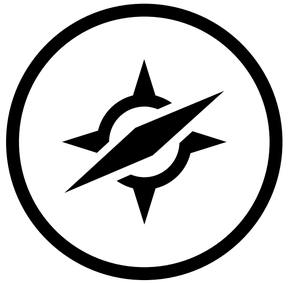
Vanilla CSS

@scope

Credit where credit is due

I'm just playing around, using what I can, and reading the same documentation and articles as y'all. So high-five to all the people who work on MDN blog elsewhere about all this stuff.





Interop 2025

Browser compatibility

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS
`@scope`	✓ 118	✓ 118	✗ 128 *	✓ 104	✓ 17.4	✓ 118	✗ No	✓ 79	✓ 17.4

web.dev

web.dev Resources More Search

The focus areas for 2025

- Anchor positioning
- `backdrop-filter`
- Core Web Vitals
- `<details>` element
- Layout
- Modules
- Navigation API
- Pointer and mouse events
- Remove mu nts
- `@scope`
- `scrollend`
- Storage Access API
- `text-decoration`
- `URLPattern`

Allow me to introduce `@scope` with a stupid usage scenario that doesn't help anything.

Then we'll learn later that it's not entirely stupid.

```html
<div class="card">
  <h3>Title</h3>
  <p>Description</p>
  <a
    href="/product/a"
    class="button">
      View Product A
  </a>
</div>
```

```css
.card {
  h3 { }
  p { }
  .button { }
}
```

```html
<div class="card">
  <h3>Title</h3>
  <p>Description</p>
  <a
    href="/product/a"
    class="button">
      View Product A
  </a>
</div>
```

```css
@scope (.card) {
  :scope { }
  h3 { }
  p { }
  .button { }
}
```

```
@supports (at-rule(@scope)) {
  @scope (.card) {
    :scope { }
    h3 { }
    p { }
    .button { }
  }
}


if (window.CSSScopeRule) {
  document.documentElement.classList.add("scope-supported");
}
```

The basic thing that is not useful with `@scope`

1. **Just scoping a single selector**

Three *niche* things that are useful about `@scope`

1. **Donut Scope**
2. **DOM Blasters**
3. **Proximity**

Donut SCOPE

The "Stop Caring Now" Selector

stubbornella.org

Nicole Sullivan

Geek Photos Resume Events About Videos

GOAT

2011

Scope donuts

Note: This article is esoteric-c
you only like to hear about wha
been warned. 😉

At first, when the HTML5 work
skeptical. I thought, "oh dear, t
cause massive duplication anc
about the tool and hoping we c

Scope is great f

The first that comes to mind, a
you want to pull in a twitter fee

stubbornella.org

What you can see is that tabs are an extension of an ordinary box:
head and body wrapped in a div with the classes `box` and `tabs`. What makes the
tabs unique is that both the head and body are filled with unordered lists which
correspond to the tab control and tab body respectively.

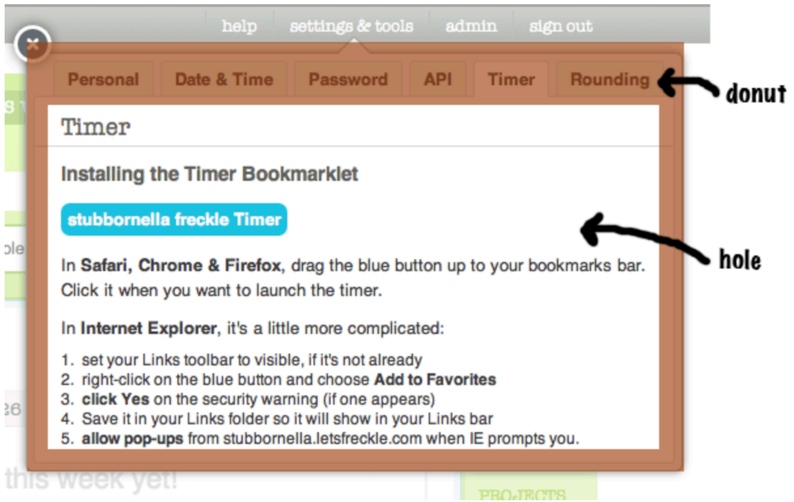


Figure 2: Tabs with the donut highlighted. We don't want any styles falling either

STEVE NADIS SCIENCE MAR 16, 2025 7:00 AM

Undergraduate Disproves 40-Year-Old Conjecture, Invents New Kind of Hash Table

A young computer scientist and two colleagues show that searches within data structures called hash tables can be much faster than previously deemed possible.

ILLUSTRATION: NASH WEERASEKERA FOR *QUANTA MAGAZINE*

THE ORIGINAL VERSION *of this story appeared in Quanta Magazine.*

Sometime in the fall of 2021, Andrew Krapivin, an undergraduate at Rutgers University, encountered a paper that would change his life. At the time, Krapivin didn't give it much thought. But two years later, when he finally set aside time to go through the paper ("just for fun," as he put it), his efforts would lead to a rethinking of a widely used tool in computer science.

Science Newsletter

Your weekly roundup of the best stories on health care, the climate crisis, new scientific discoveries, and more. Delivered on Wednesdays.

SIGN UP

By signing up, you agree to our user agreement (including class action waiver and arbitration provisions), and acknowledge our privacy policy.

The paper's title, "Tiny Pointers," referred to arrowlike entities that can direct you to a piece of information, or element, in a computer's memory. Krapivin soon came up with a potential way to further miniaturize the pointers so they consumed less memory. However, to achieve that, he needed a better way of organizing the data that the pointers would point to.

MOST POPULAR

GEAR
Nearly All Cybertrucks Have Been Recalled Because Tesla Used the Wrong Glue
BY AARIAN MARSHALL

SECURITY NEWS
Low-Cost Drone Add-Ons From China Let Anyone With a Credit Card Turn Toys Into Weapons of War
BY LILY HAY NEWMAN

SCIENCE
Undergraduate Disproves 40-Year-Old Conjecture, Invents New Kind of Hash Table
BY STEVE NADIS

GEAR NEWS AND EVENTS
DOGE's Dodgy Numbers Employ a

Underlines in body copy is good.

LUSTRATION: NASH WEERASEKERA FOR *QUANTA MAGAZINE*

THE ORIGINAL VERSION *of this story appeared in Quanta Magazine.*

Sometime in the fall of 2021, Andrew Krapivin, an undergraduate at Rutgers University, encountered a paper that would change his life. At the time, Krapivin didn't give it much thought. But two years later, when he finally set aside time to go through the paper ("just for fun," as he put it), his efforts would lead to a rethinking of a widely used tool in computer science.

Science Newsletter

Your weekly roundup of the best stories on health care, the climate crisis, new scientific discoveries, and more. Delivered on Wednesdays.

SIGN UP

By signing up, you agree to our user agreement (including class action waiver and arbitration provisions), and acknowledge our privacy policy.

The paper's title, "Tiny Pointers," referred to arrowlike entities that can direct you to a piece of information, or element, in a computer's memory. Krapivin soon came up with a potential way to further miniaturize the pointers so they consumed less memory. However, to achieve that, he needed a better way of organizing the data that the pointers would point to.

MOST POPULAR

Don't need the underline in other areas.

MOST POPULAR

GEAR

Nearly All Cybertrucks Hav
Recalled Because Tesla Use
Wrong Glue

BY AARIAN MARSHALL

SECURITY NEWS

Low-Cost Drone Add-Ons
China Let Anyone With a C
Card Turn Toys In Weapo
War

BY LILY HAY NEWMAN

SCIENCE

Undergraduate Disproves 4
Old Conjecture, Invents Ne
of Hash Table

BY STEVE NADIS

GEAR NEWS AND EVENTS

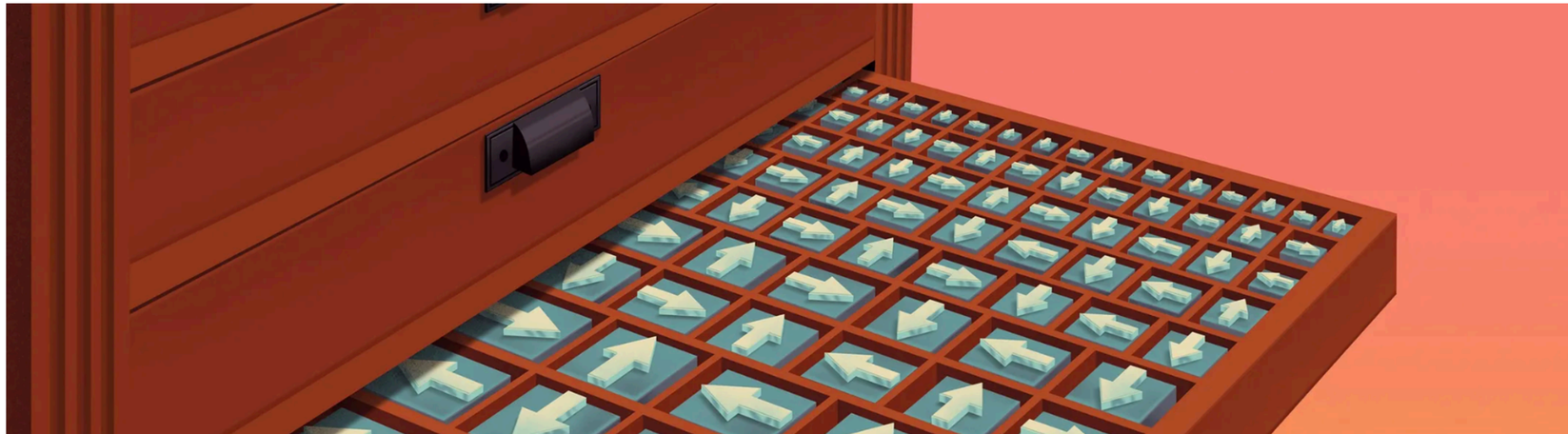
DOGE's Dodgy Numbers En

WIRED SECURITY POLITICS GEAR THE BIG STORY BUSINI

STEVE NADIS SCIENCE MAR 16, 2025 7:00 AM

Undergraduate Disproves 40-Ye e, Invents

A young computer scientist and their colleagues show that searches within data structures called hash possible.



```
/* links are underlined
   in user-agent stylesheet */

.site-nav {
  a {
    text-decoration: none;
  }
}

.popular-articles {
  h2 a {
    text-decoration: none;
  }
}
```

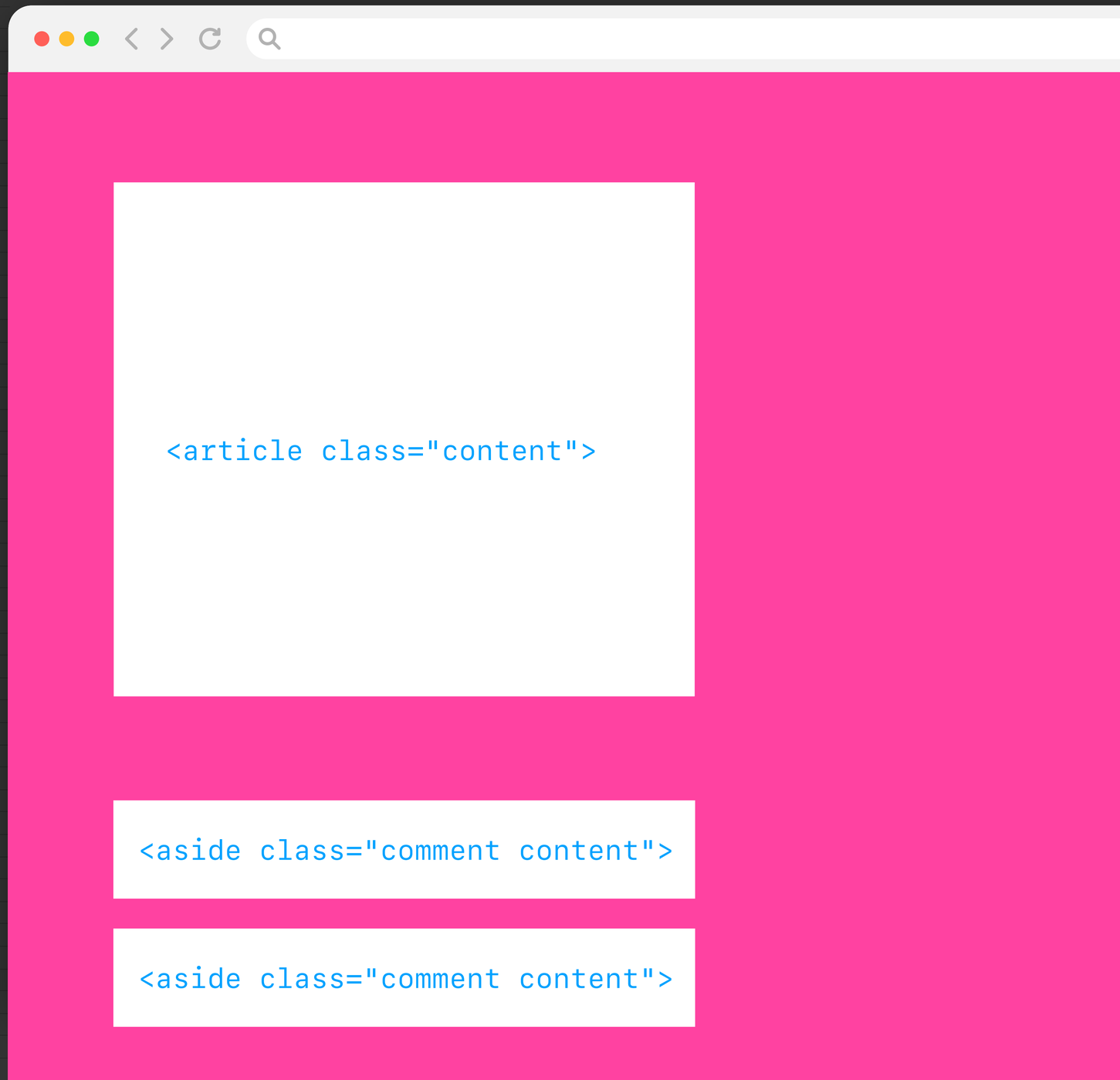
```
a {
  text-decoration: none;
}
.content {
  a {
    text-decoration: revert;
  }
}
```

Remove underline from links everywhere... until you get to any element with a class of content, then stop doing that.

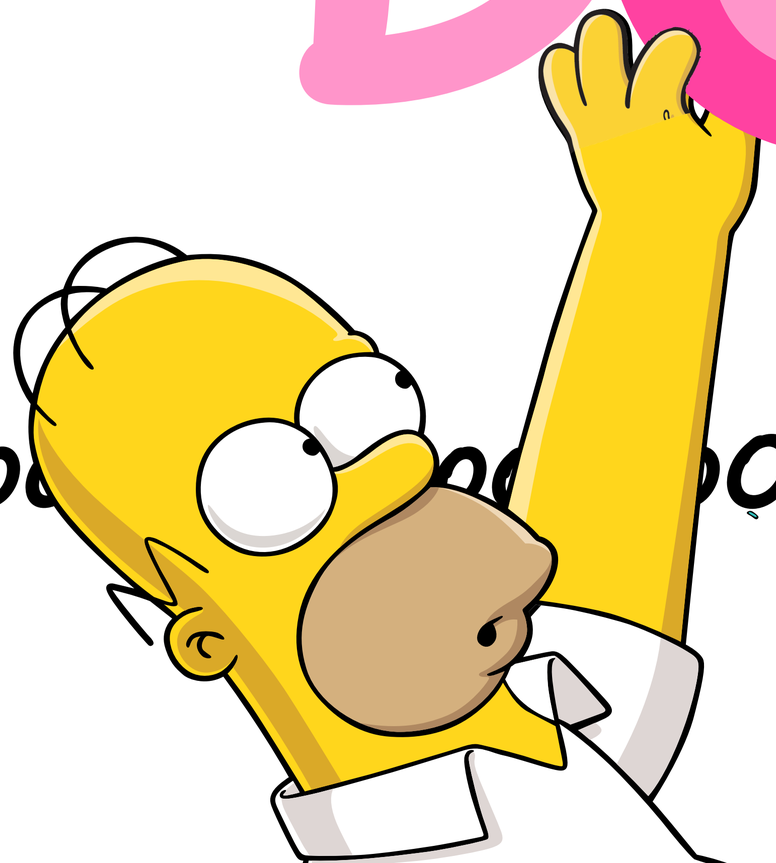
```
@scope (body) to (.content) {
  a {
    text-decoration: none;
  }
}
```

Donut

```
@scope (body) to (.content) {
  a {
    text-decoration: none;
  }
}
```



Donut



oooooooooooooOOOooOOooooo lexical scoping

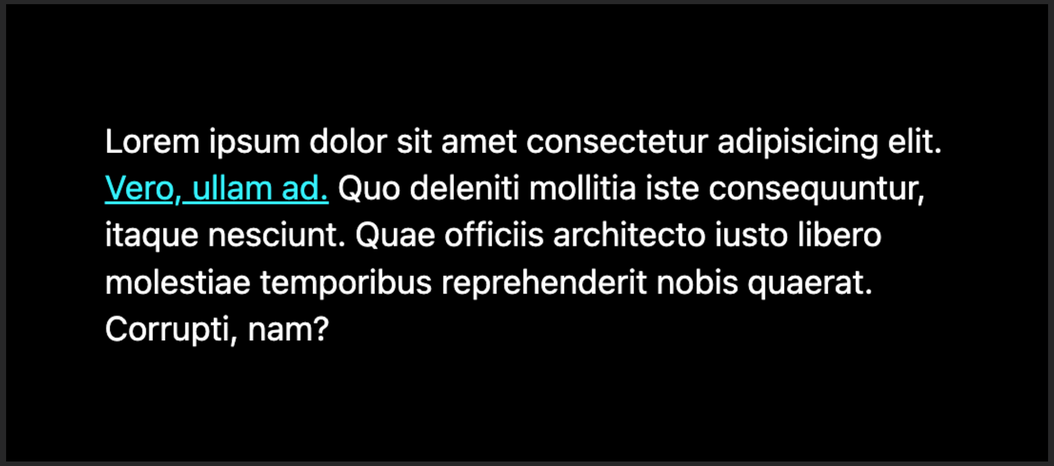
Proximity

Light Mode / Dark Mode has become the ultimate classic demo for this, and I'm not above it, so let's do it. We'll do another one too.

```html
<!-- or theme-dark -->
<main class="theme-light">
  <p>
    Lorem ipsum dolor sit amet consectetur
    adipisicing elit. <a href="#blah">Vero,
    ullam ad.</a> Quo deleniti mollitia iste
    consequuntur, itaque nesciunt. Quae
    officiis architecto iusto libero molestiae
    temporibus reprehenderit nobis quaerat.
    Corrupti, nam?
  </p>
</main>
```

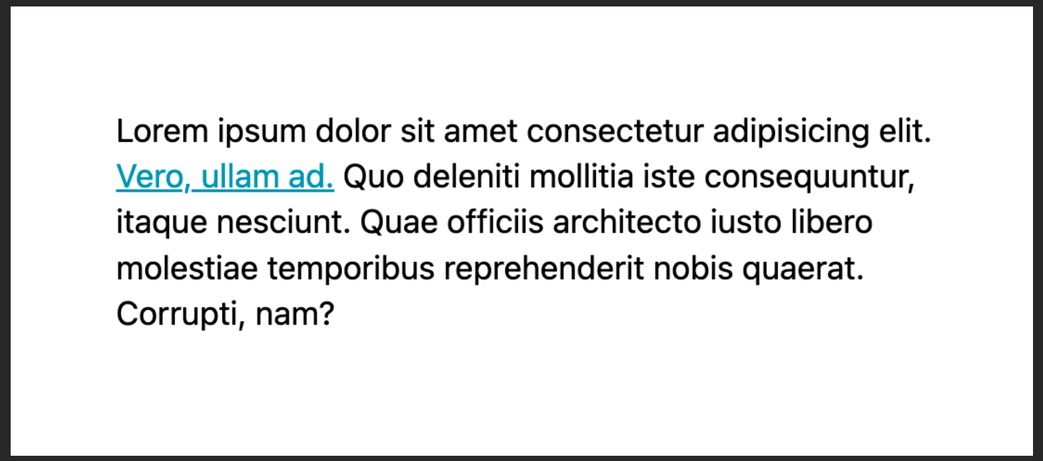
```
.theme-dark {
  background: black;
  color: white;

  a {
    color: oklch(0.9 0.15 212);
  }
}
```

lighter




```
.theme-light {
  background: white;
  color: black;

  a {
    color: oklch(0.6 0.15 212);
  }
}
```

darker



```html
<main class="theme-dark">
  <p>
   Lorem ipsum dolor sit amet consectetur
   adipisicing elit. <a href="#blah">Vero, ullam
   ad.</a> Quo deleniti mollitia iste
   consequuntur, itaque nesciunt. Quae officiis
   architecto iusto libero molestiae temporibus
   reprehenderit nobis quaerat. Corrupti, nam?
  </p>
  <footer class="theme-light">
    <a href="">Link</a>
    <a href="">Link</a>
    <a href="">Link</a>
  </footer>
</main>
```

theme within theme

```
.theme-dark {
  background: black;
  color: white;

  a {
    color: oklch(0.9 0.15 21
  }
}

.theme-light {
  background: white;
  color: black;

  a {
    color: oklch(0.6 0.15 212);
  }
}
```

Source order matters.

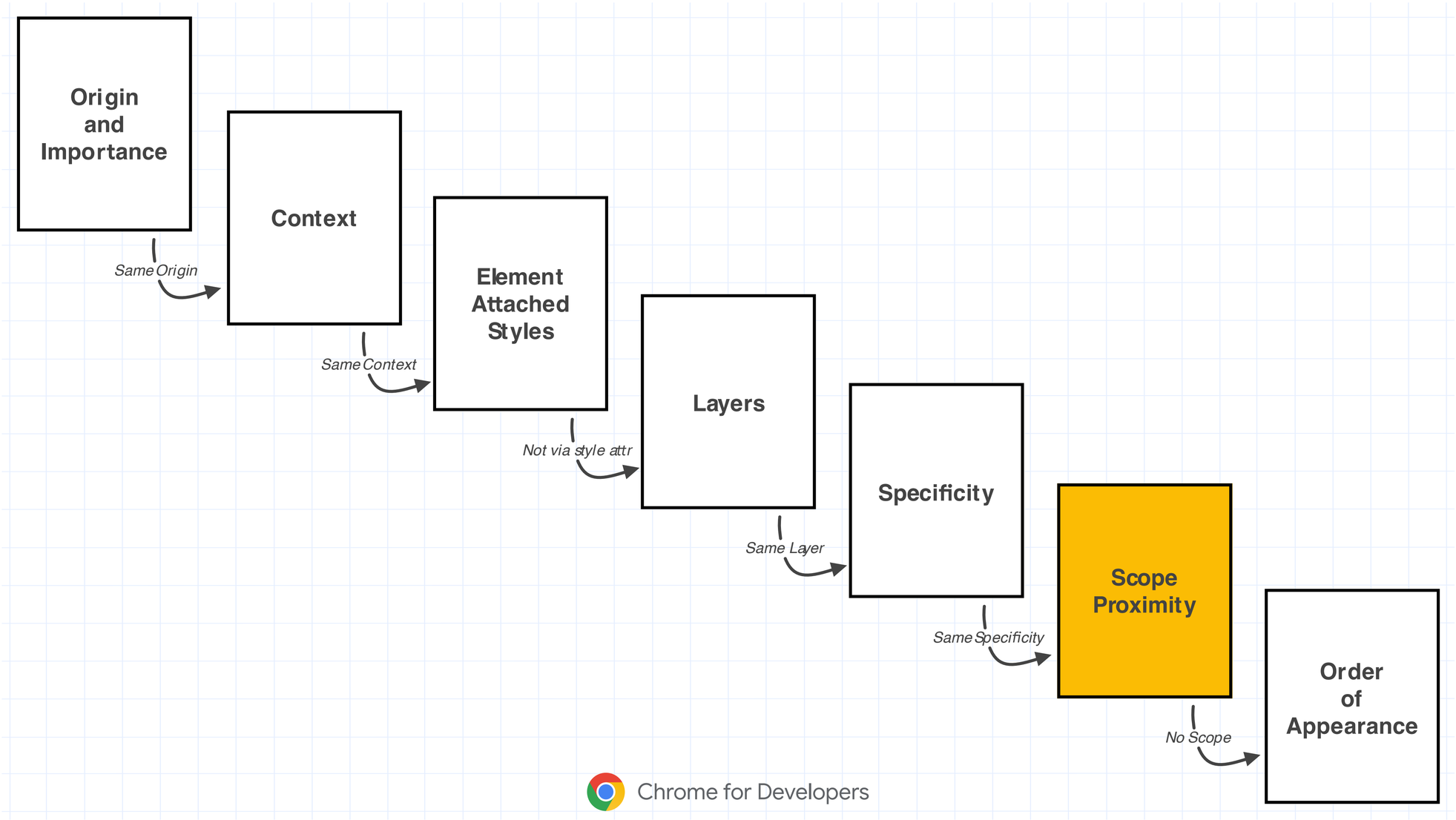
Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, ullam ad. Quo deleniti mollitia iste consequuntur, itaque nesciunt. Quae officiis architecto iusto libero molestiae temporibus reprehenderit nobis quaerat. Corrupti, nam?

Link Link Link

Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, ullam ad. Quo deleniti mollitia iste consequuntur, itaque nesciunt. Quae officiis architecto iusto libero molestiae temporibus reprehenderit nobis quaerat. Corrupti, nam?

Link Link Link

```
@scope (.theme-dark) {
  background: black;
  color: white;

  a {
    color: oklch(0.9 0.15 212);
  }
}

@scope (.theme-light) {
  background: white;
  color: black;

  a {
    color: oklch(0.6 0.15 212);
  }
}
```



https://developer.chrome.com/docs/css-ui/at-scope

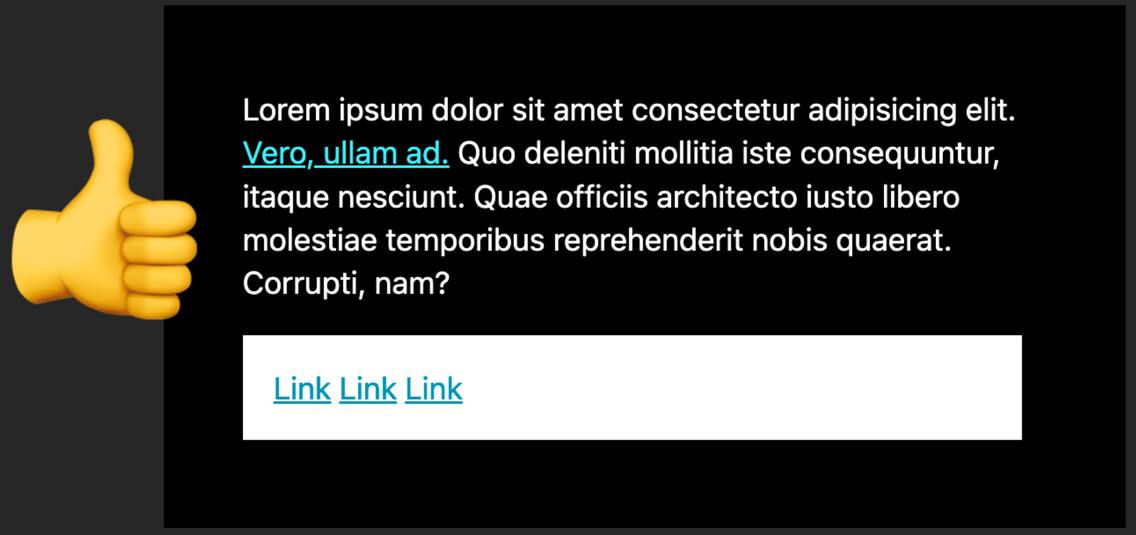
For the links...

```
<main class="theme-dark">
  <p>
   Lorem ipsum dolor sit amet consectetur
   adipisicing elit. <a href="#blah">Vero, ullam
   ad.</a> Quo deleniti mollitia iste
   consequuntur, itaque nesciunt. Quae officiis
   architecto iusto libero molestiae temporibus
   reprehenderit nobis quaerat. Corrupti, nam?
  </p>
  <footer class="theme-light">
    <a href="">Link</a>
    <a href="">Link</a>
    <a href="">Link</a>
  </footer>
</main>
```

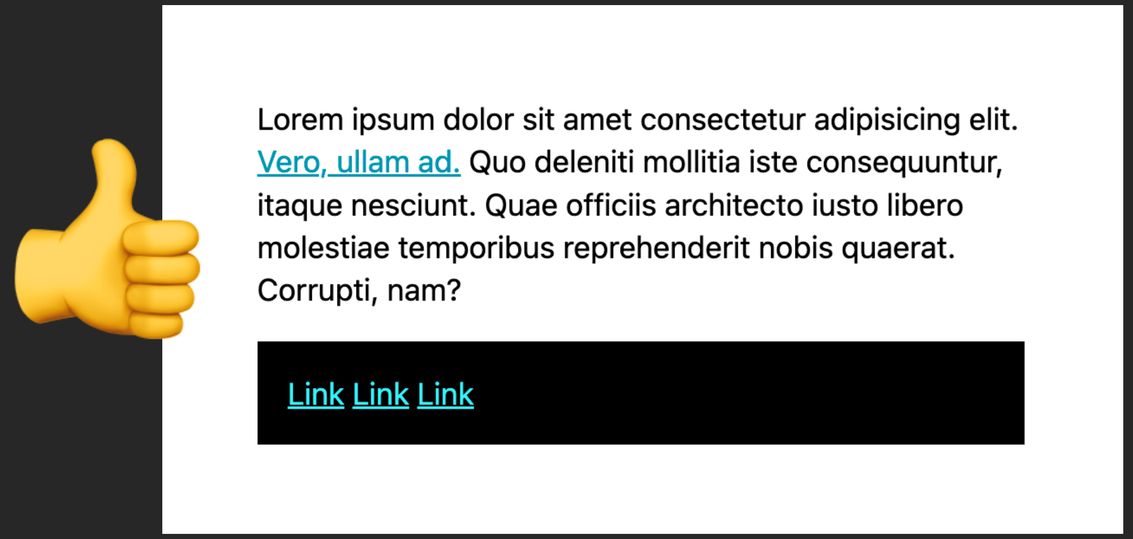
This class has lower proximity.

This class has higher proximity.

```
@scope (.theme-dark) {
  background: black;
  color: white;

  a {
    color: oklch(0.9 0.15 212);
  }
}
```




```
@scope (.theme-light) {
  background: white;
  color: black;

  a {
    color: oklch(0.6 0.15 212);
  }
}
```



ok class — what did we learn?

`@scope` can help reduce concerns about source order.

```html
<div class="card">
  <img />
  <h2>I'm a card</h2>
</div>

<div class="card card-big">
  <img />
  <h2>I'm a card</h2>
</div>
```

```css
.card-big {
  grid-column: span 2;
  display: flex;

  img {
    width: 50%;
    height: 100%;
  }
}

.card {
  img {
    width: 100%;
    aspect-ratio: 16 / 9;
    object-fit: cover;
  }
}
```

```html
<div class="card">
  <img />
  <h2>I'm a card</h2>
</div>

<div class="card card-big">
  <img />
  <h2>I'm a card</h2>
</div>
```

Keeps specificity equal

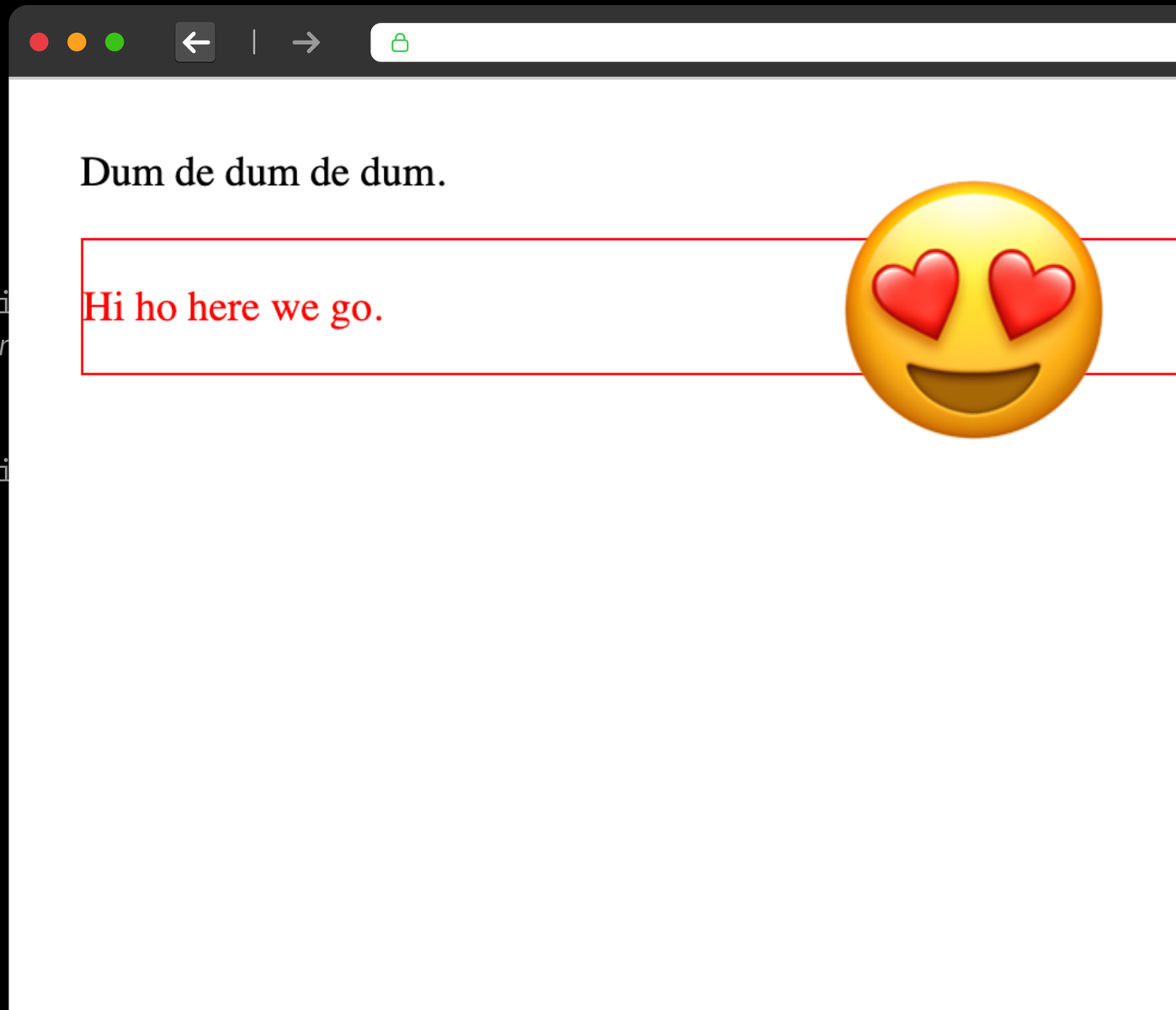
```css
@scope (.card-big) {
  :scope {
    grid-column: span 2;
    display: flex;

    img {
      width: 50%;
      height: 100%;
    }
  }
}

.card {
  img {
    width: 100%;
    aspect-ratio: 16 / 9;
    object-fit: cover;
  }
}
```

DOM Blasters

One-offs FTW

```
<main>

  <div>
    <p>Dum de dum de dum.<p>
  </div>

  <div>
    <p>Hi ho here we go.</p>

    <style>
     @scope { /* Scope is the <div> above, as this is a direct child. */
       :scope { /* This selects the <div>, optional. */
         border: 1px solid red;

         /* I can use CSS nesting in here, ensuring *everything* is safely scoped */
         p {
           color: red;
         }
       }
     }
    </style>

  </div>

</main>
```

```
<main>

  <div>
    <p>Dum de dum de dum.<p>
  </div>

  <div>
    <p>Hi ho here we go.</p>

    <style>
     @scope { /* Scope is the <div> above, as thi
       :scope { /* This selects the <div>, option
         border: 1px solid red;

         /* I can use CSS nesting in here, ensuri
         p {
           color: red;
         }
       }
     }
    </style>

  </div>

</main>
```



This... is... basically selector scoping?

Without the selectors? The parent, anyway.

Just put the `<style>` where you wanna scope.

Wow?

Kinda like Shadow DOM without using the Shadow DOM.

Zero tooling native web platform component scoping.

But there are **unknowns** here for me.

Is it performant to have every single component on a site have a `<style>` tag plopped inside of it? Is there DOM weight or memory problems with that? Is it not taking advantage of the power of cache? Would HTML streaming help?

es-4605262320250601-01971d3b-69d6-7047-9747-62931d7f7c30.codepen.dev

Normal CSS Approach @scope approach

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

Premium Experience

Crafted with the same attention to detail that makes Apple products extraordinary.

- Seamless integration
- Intuitive Design
- Premium Materials
- Expecptional Performance

Get Started

1,000 Cards

	Normal CSS	Inline @scope
HTML	12,000 lines 3.0 kB transferred 472 kB resource	150,000 lines 18 kB transferred 3,176 kB resource
LCP (20x CPU Slowdown kept parity)	1 s	1 s
Lighthouse Performance	100	100
Memory	150 MB	150 MB

Low Effort Art Direction

My Cool Blog Post.

Publish

HTML

HTML Preview

My Cool Blog Post.

Let's all take a nap.

```
<style>
  @scope {
    p {
      font-size: 70px;
      color: red;
      line-height: 1.1;
    }
  }
</style>
```

And then another one.

Search 20 years of drivel...

MAY 16, 2025

My Cool Blog Post.

Let’s all take a nap.

And then another one.

Related

Attribute Superpowers Taken All The Way
10/26/2023
Similar post

Style Scoped
10/19/2023
With 2 comments

Modern CSS in Real Life
06/06/2023
With 8 comments

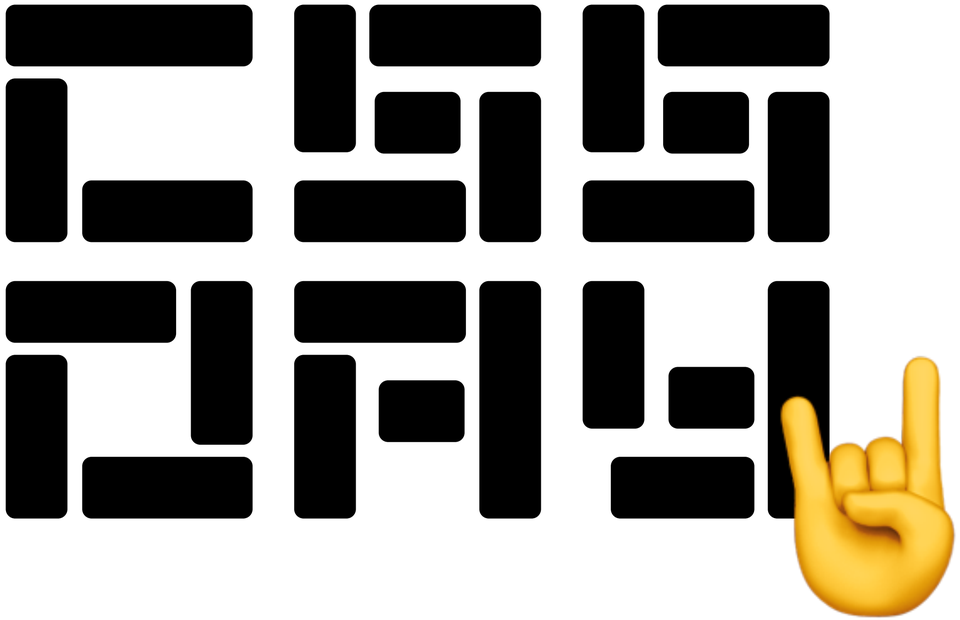
This is as close to native CSS helping with selector scoping as we've got. It's just as good as `<style scoped>`, and I'm tempted to say as good as we'll ever get.

ok let's be **done.**

The Takeaways

- CSS is naturally scoped. That might be all you need.
- Selector scoping is pretty sweet on large sites with teams. Lots of existing tools help with selector scoping, or avoid the need for it.
- Native CSS @scope is nice to have, I guess. Donut scoping and proximity are a small upgrade to CSS.
- <style> block @scope might be great. Tool-free component scoping. Can we make it performant?

Thank you!



Also, subscribe to CodePen PRO, obviously.