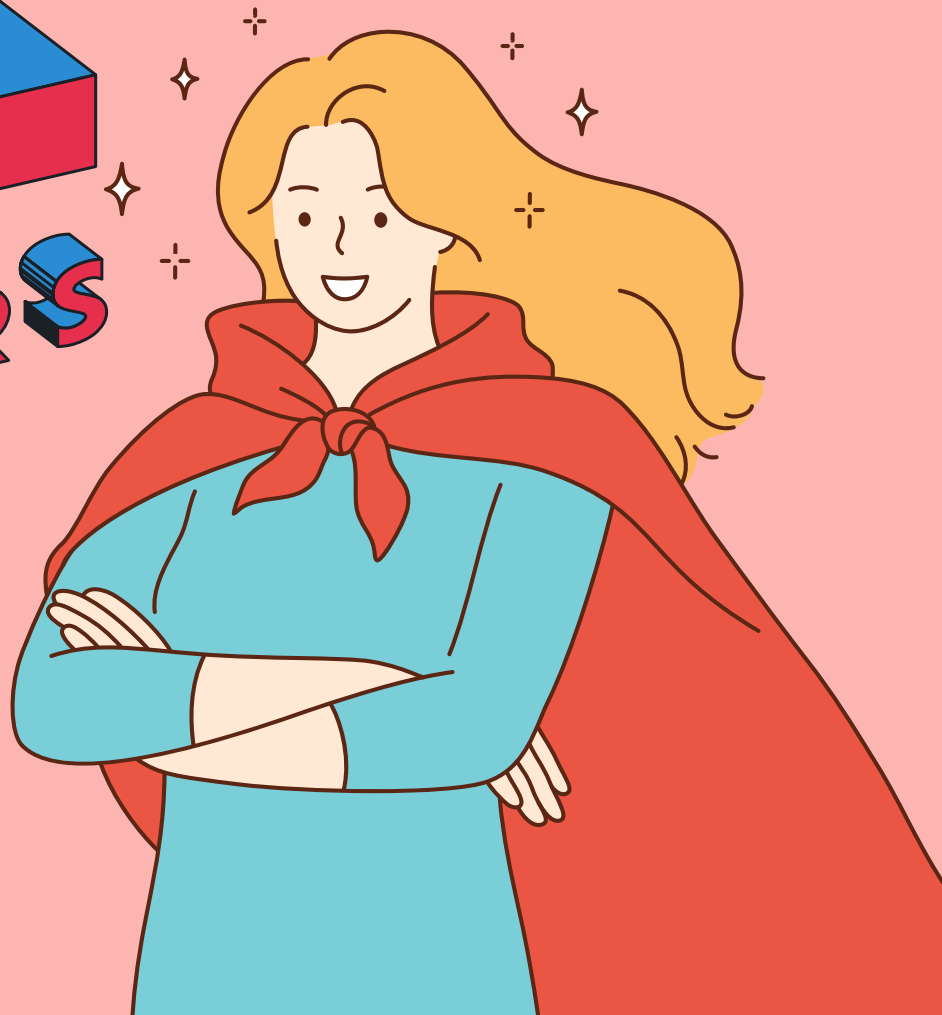
HTML

WITH SUPERPOWERS



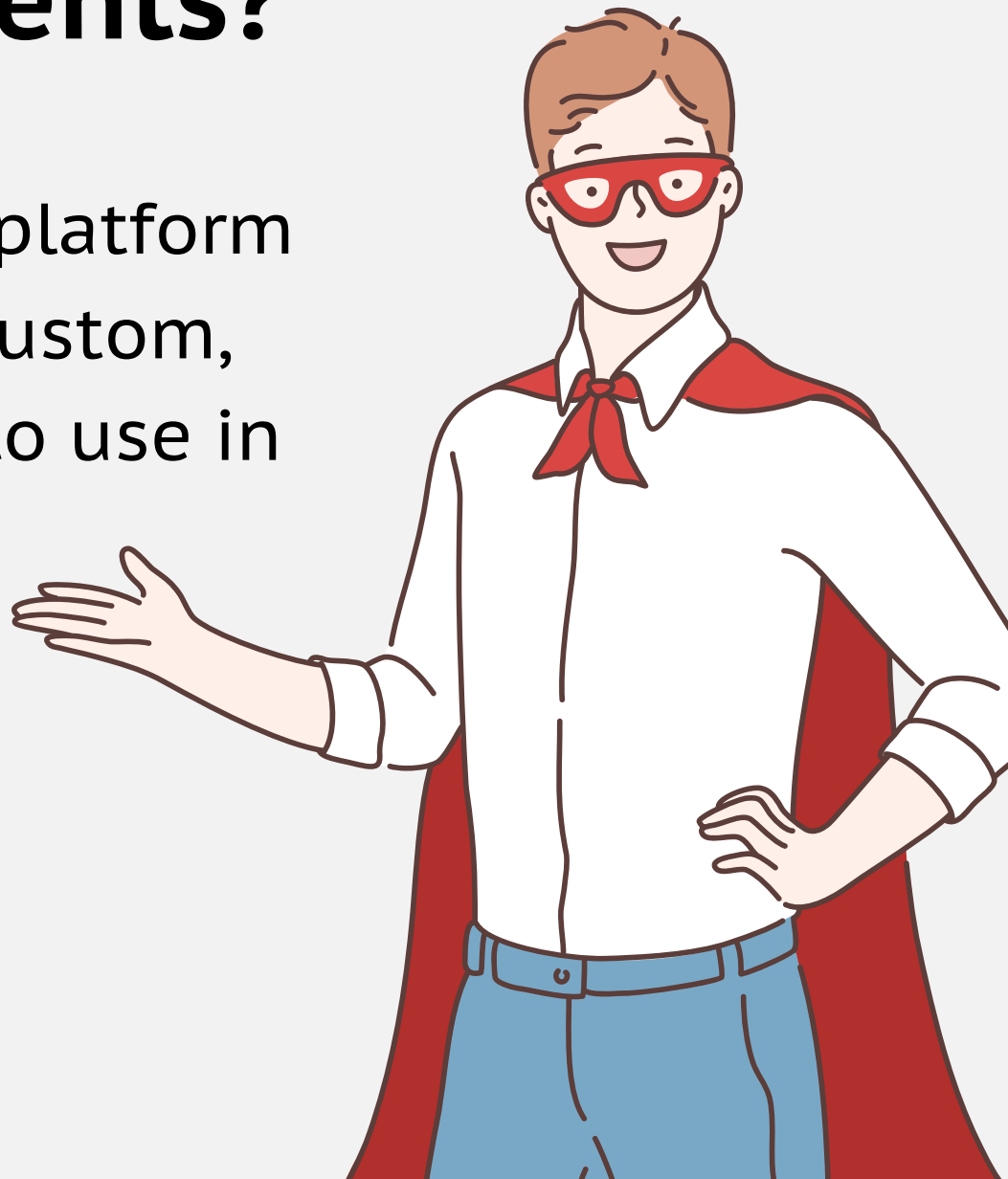
Howdy, I’m Dave Rupert

Work at Paravel in Austin, TX
Co-host ShopTalk podcast with Chris Coyier
W3C Web Components Community Group (member)

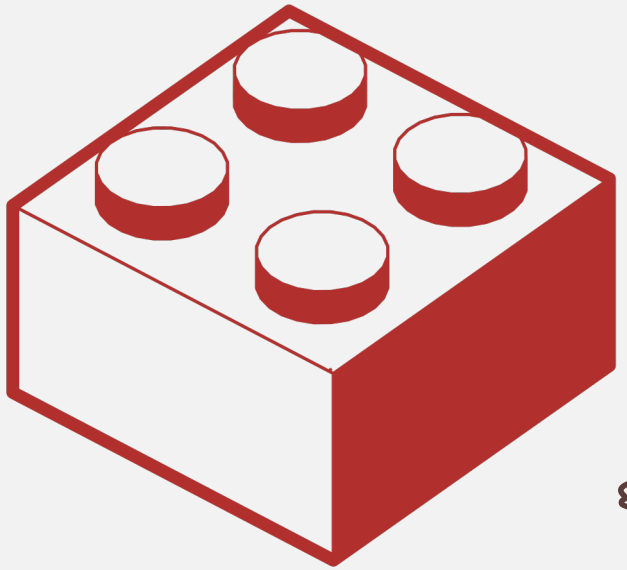
What are Web Components?

“Web components are a set of web platform APIs that allow you to create new custom, reusable, encapsulated HTML tags to use in web pages and web apps.”

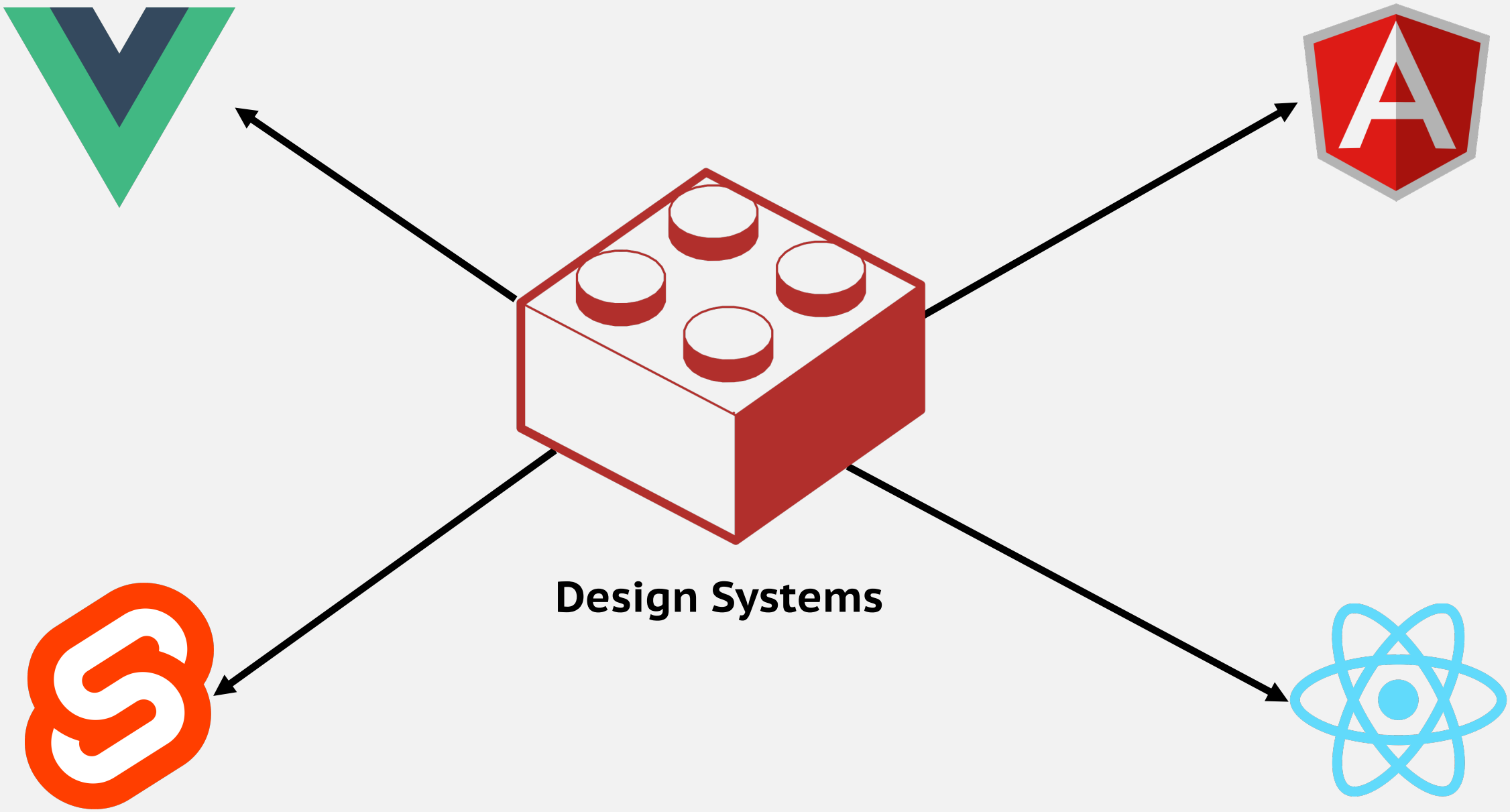
– webcomponents.org



What are Web Components?



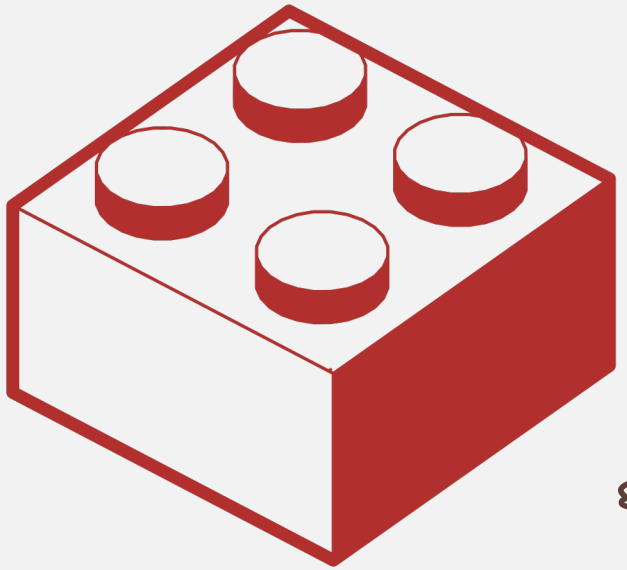
Created by Lluisa Iborra
from Noun Project


Design Systems

A Brief, Incomplete History of Web-based Component Systems

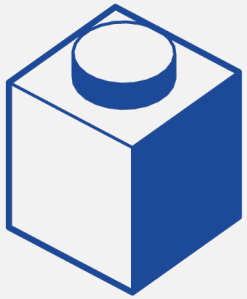
2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016

What makes Web Components so unique?

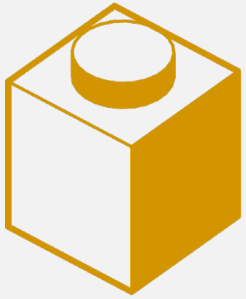


Created by Lluisa Iborra
from Noun Project

4 native web technologies

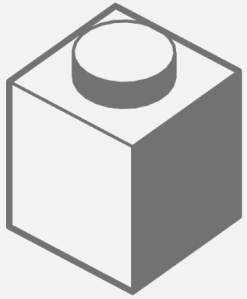


HTML <template>



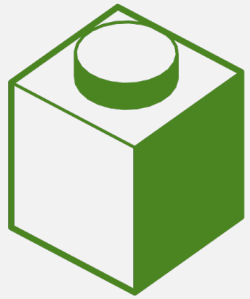
Custom Elements

<custom-element>



~~Shadow DOM~~

"Encapsulation"



~~HTML Imports~~

ES Modules

Created by Lluisa Iborra
from Noun Project

Great. But...

Are any companies actually using this?

[social validation slide]

According to Google…

12% of pages loaded in Chrome using Web Components

If they’re so great…

Why am I not using them?!?

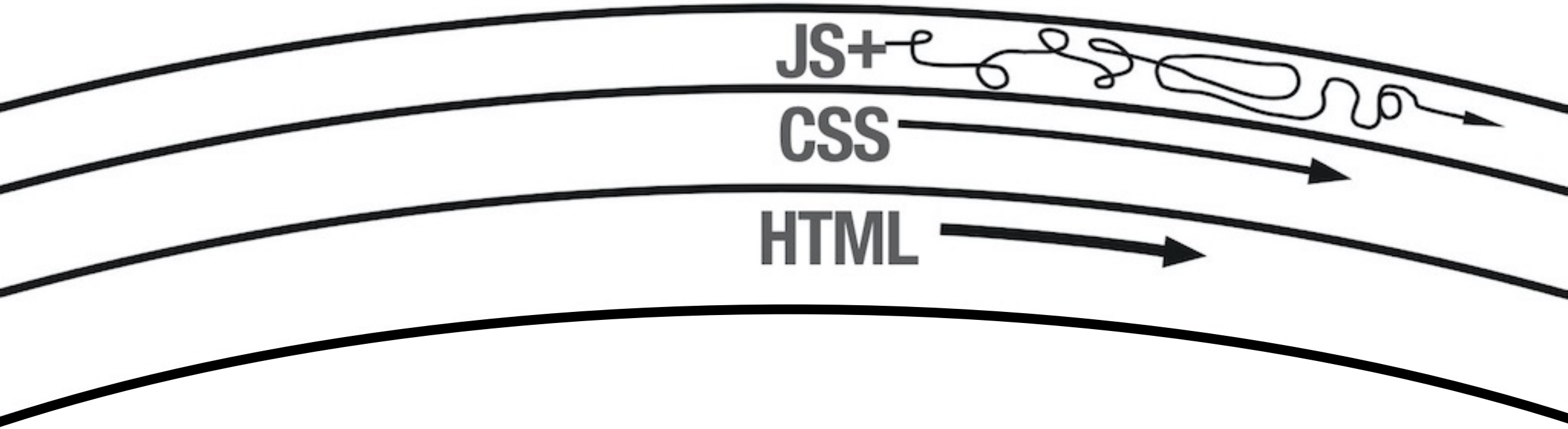
Perfectly valid question!

1. Too low-level, designed for framework authors
2. Marketing problem from heavy-handed advocacy
3. Hard to invest in something unknown
4. Not truly supported until very recently

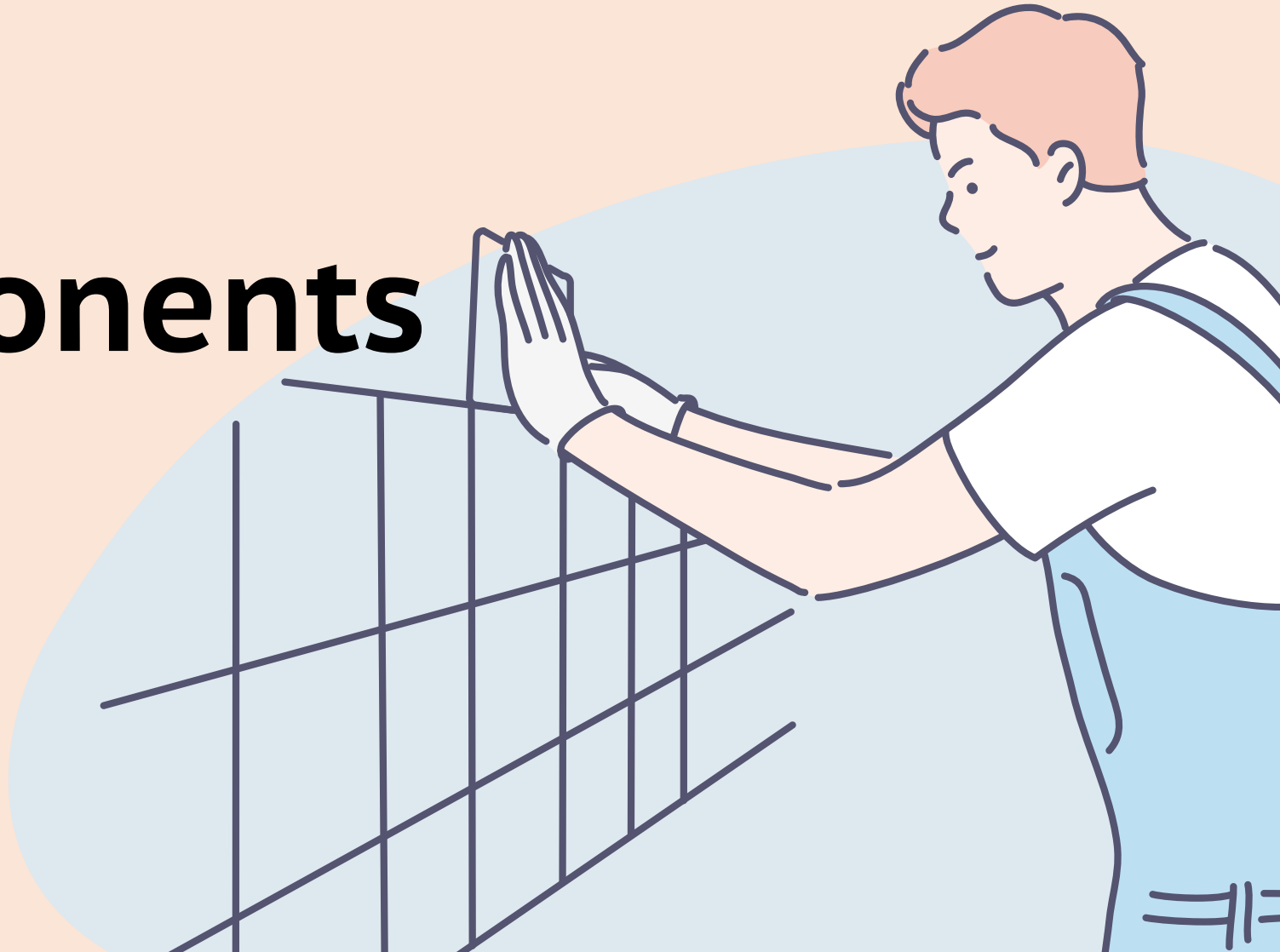
Slow like brisket...

Web Components will always be slower than HotNewFramework.js because they are Web Standards, but that slowness is what can make them a stable foundation to build on.

BUILDING


JS+
CSS
HTML

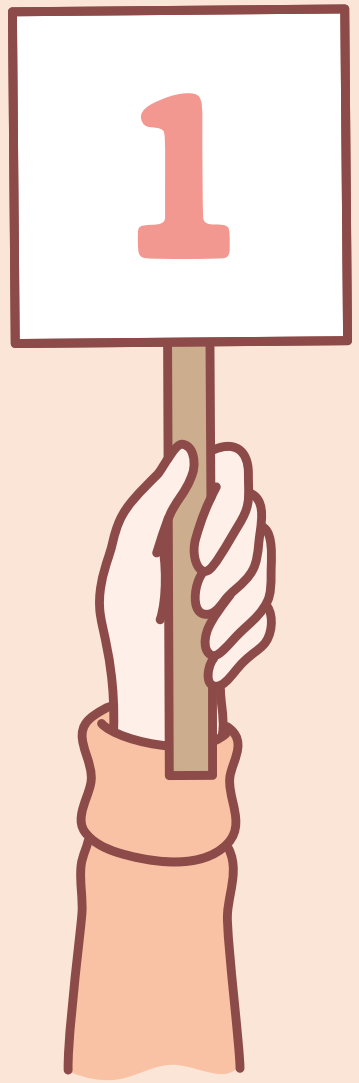
Using Web Components with HTML




JavaScript
People who just want
to write HTML and CSS

Assertion #1

Web Components bring us one step closer to writing HTML again.



Photography then and now
Japan disaster

Japan's 2011 tsunami, then and now - in pictures

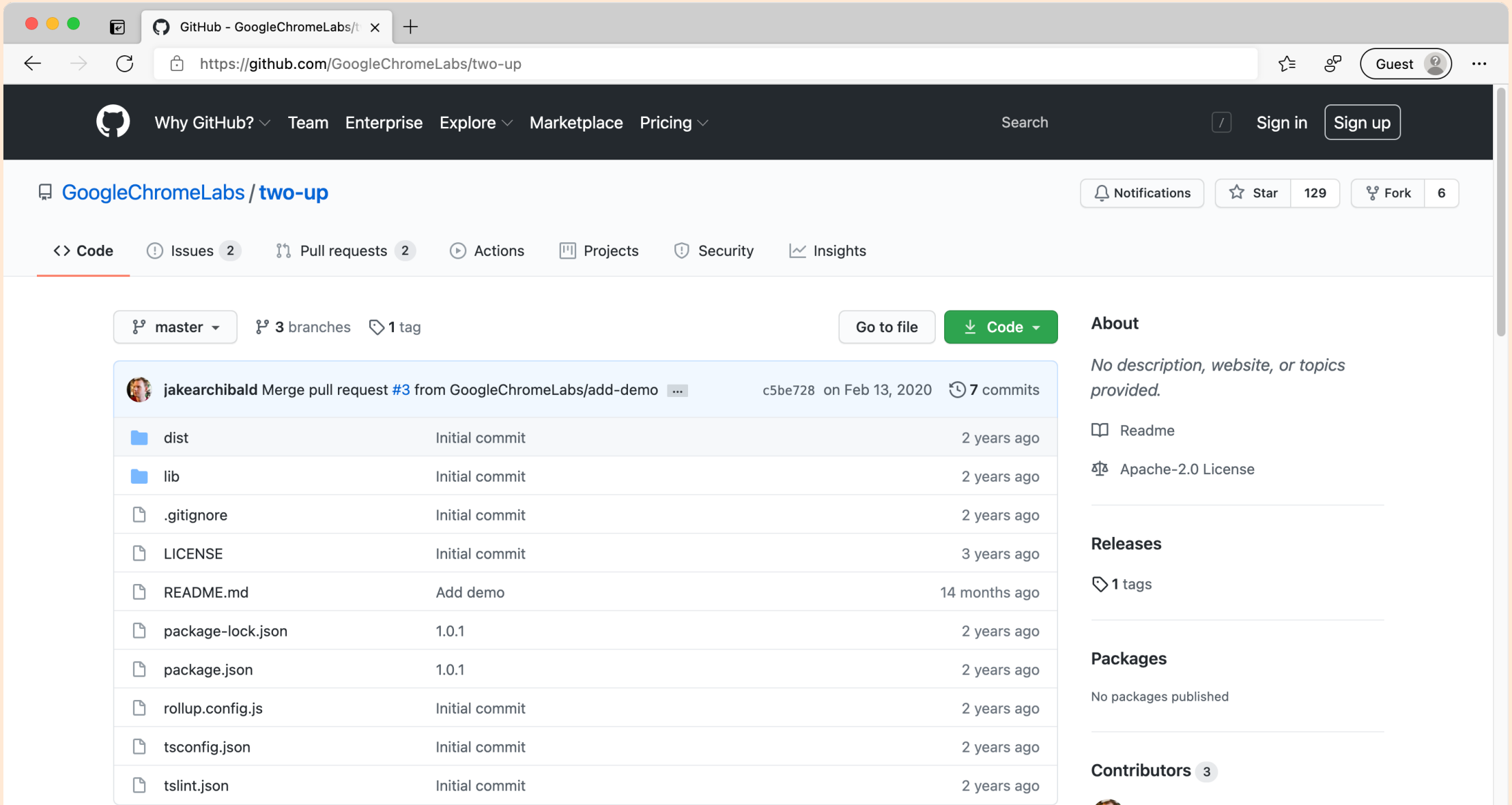
Then and now photographs show the extent of the destruction of the 2011 tsunami in Japan, and the enormity of the reconstruction work

Kazuhiro Nogi/AFP/Getty Images

Wed 10 Mar 2021 01.00 EST

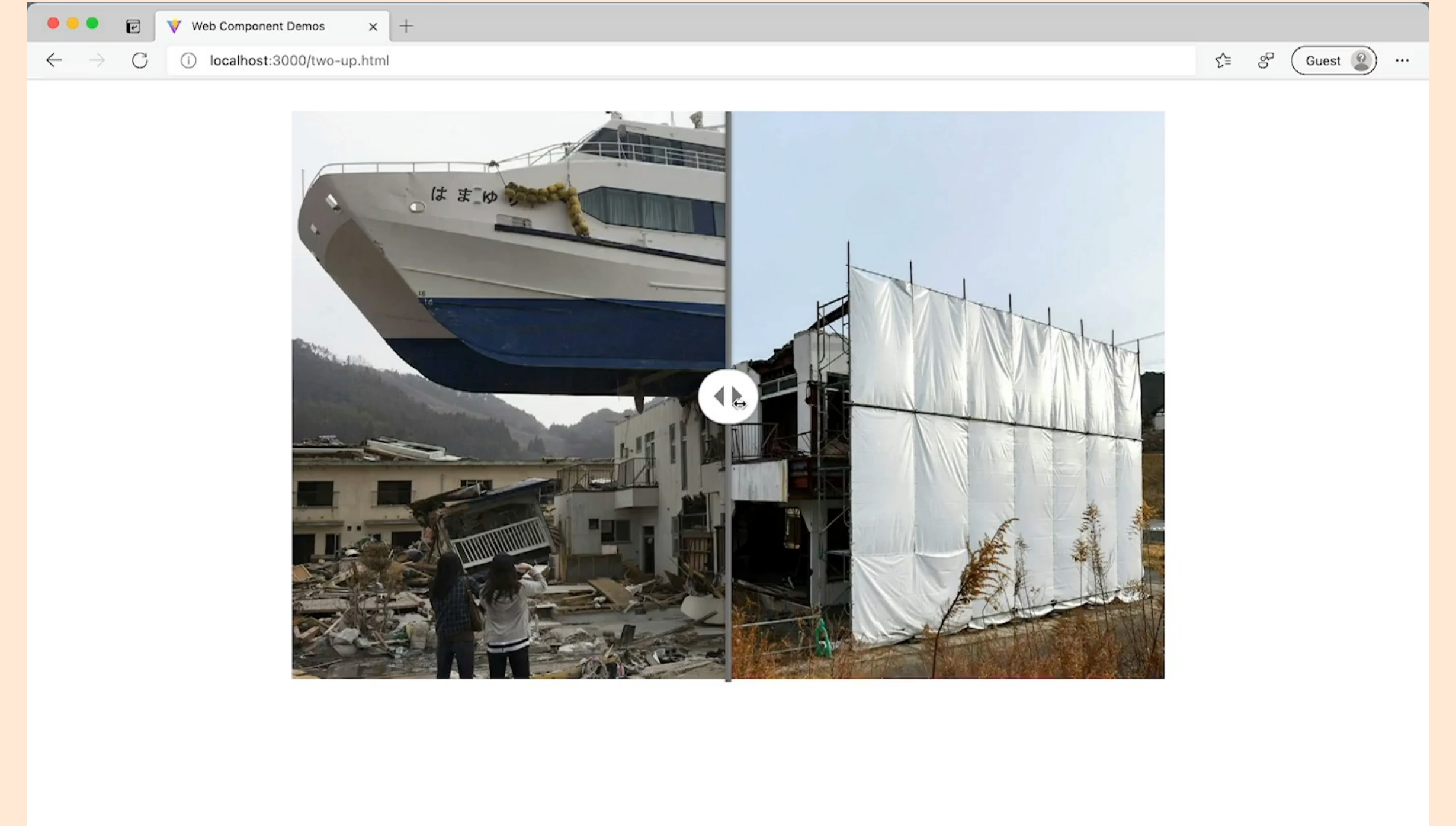
Ten years ago, one of the most powerful earthquakes on record triggered a devastating tsunami in Japan, killing more than 18,000 people and triggering catastrophic meltdowns at the Fukushima Daiichi nuclear plant. Kazuhiro Nogi's photographs compare the destruction of 2011 with the same locations following their reconstruction

<two-up>


GitHub - GoogleChromeLabs/t
https://github.com/GoogleChromeLabs/two-up
Guest
Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
GoogleChromeLabs / two-up
Notifications
Star 129
Fork 6
Code
Issues 2
Pull requests 2
Actions
Projects
Security
Insights
master
3 branches
1 tag
Go to file
Code
jakearchibald Merge pull request #3 from GoogleChromeLabs/add-demo ...
c5be728 on Feb 13, 2020
7 commits
dist Initial commit 2 years ago
lib Initial commit 2 years ago
.gitignore Initial commit 2 years ago
LICENSE Initial commit 3 years ago
README.md Add demo 14 months ago
package-lock.json 1.0.1 2 years ago
package.json 1.0.1 2 years ago
rollup.config.js Initial commit 2 years ago
tsconfig.json Initial commit 2 years ago
tslint.json Initial commit 2 years ago
About
No description, website, or topics provided.
Readme
Apache-2.0 License
Releases
1 tags
Packages
No packages published
Contributors 3

<two-up>

```
<two-up>
    <div><img src="before.jpg" alt=""></div>
    <div><img src="after.jpg" alt=""></div>
</two-up>

<script src="path/to/two-up.js"></script>
```


Web Component Demos
localhost:3000/two-up.html
Guest

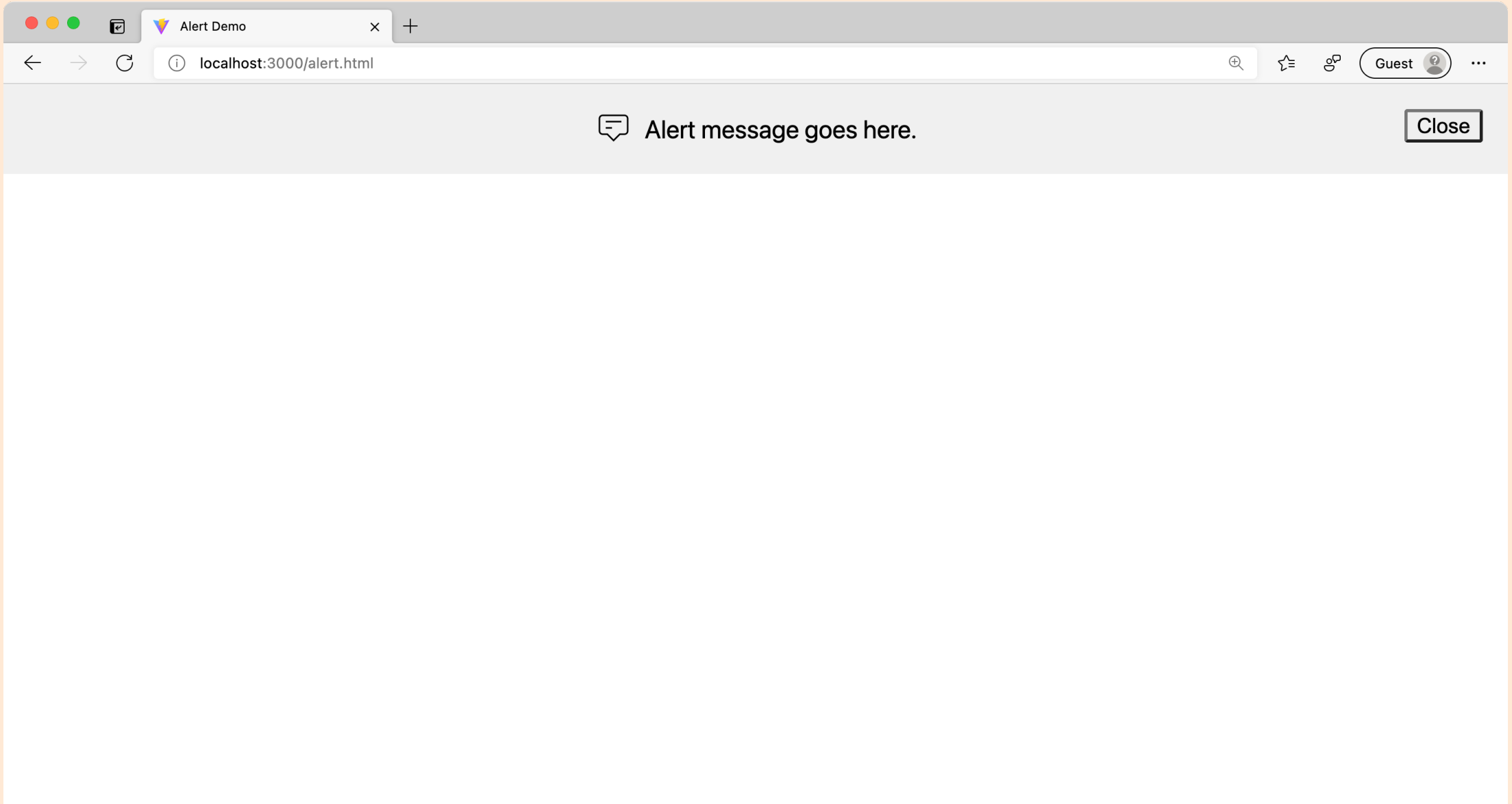
A basic example

```
<div class="alert">
    <img src="path/to/icon_note.svg" alt="" />
    <p>Alert message goes here.</p>
    <button>Close</button>
</div>
```

A basic example

```
<custom-alert icon="note">
    <p>Alert message goes here.</p>
</custom-alert>

<script src="path/to/custom-alert.js"></script>
```

A basic example

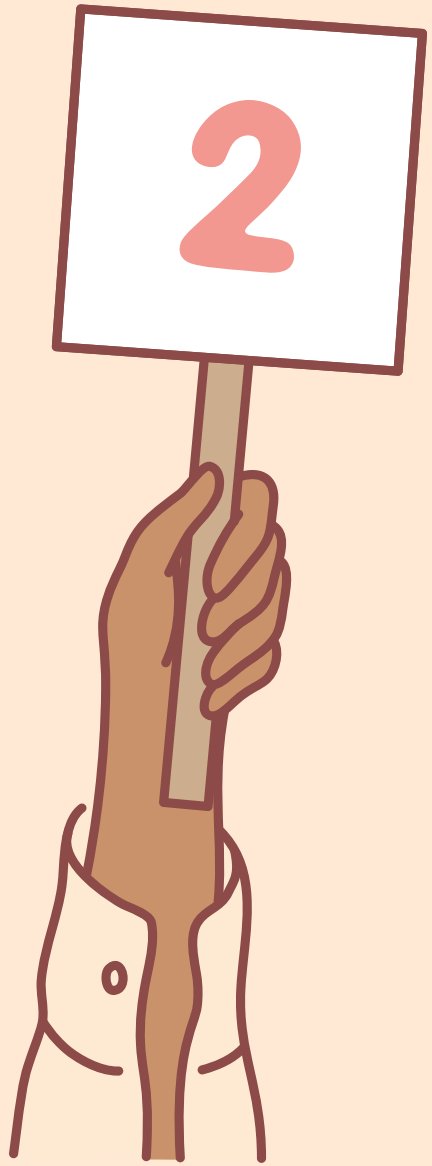


A basic example

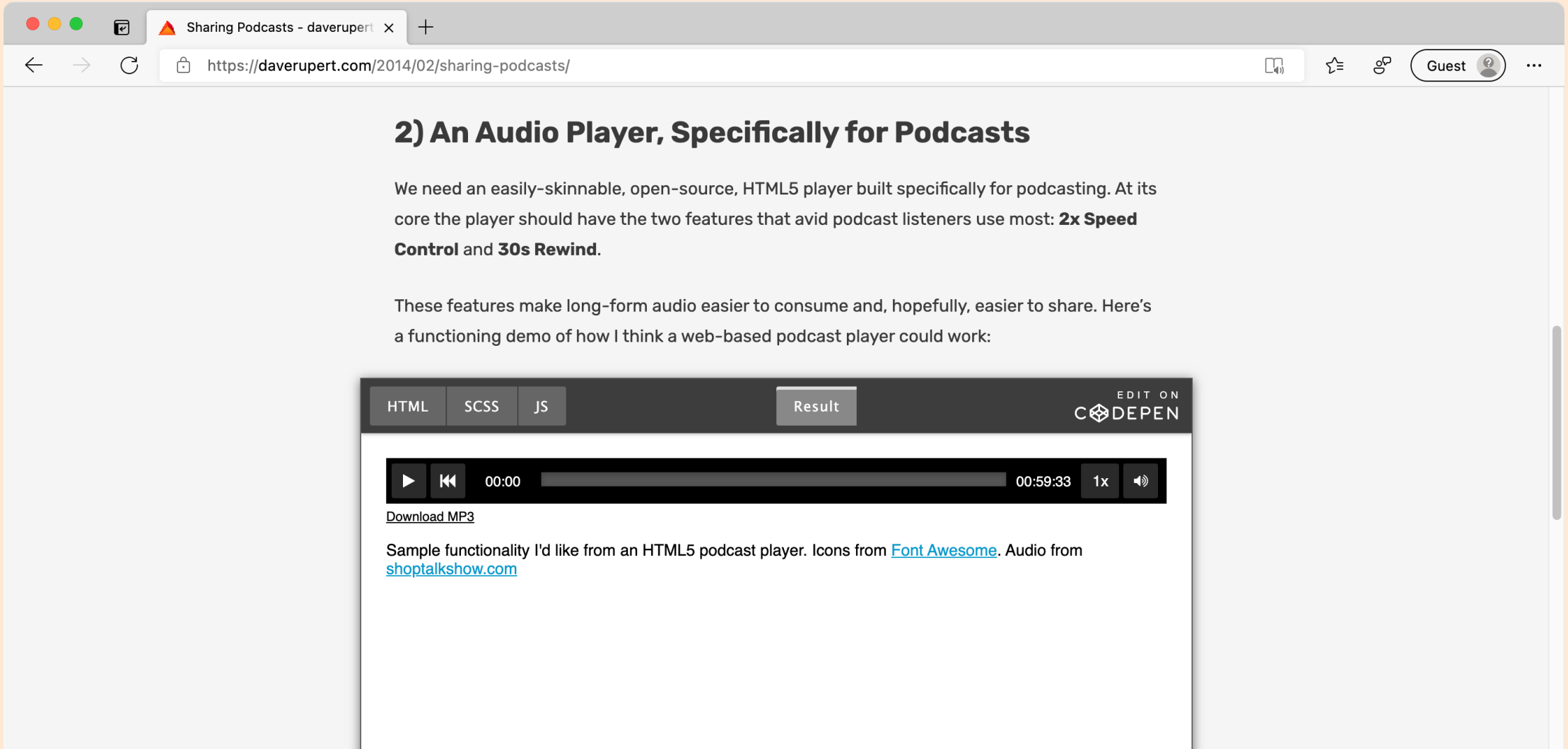
```
<custom-alert icon="note">
    <p>Alert message goes here.</p>

    <img src="path/to/icon_note.svg" />
    <slot></slot>
    <button>Close</button>

</custom-alert>
```

Assertion #2

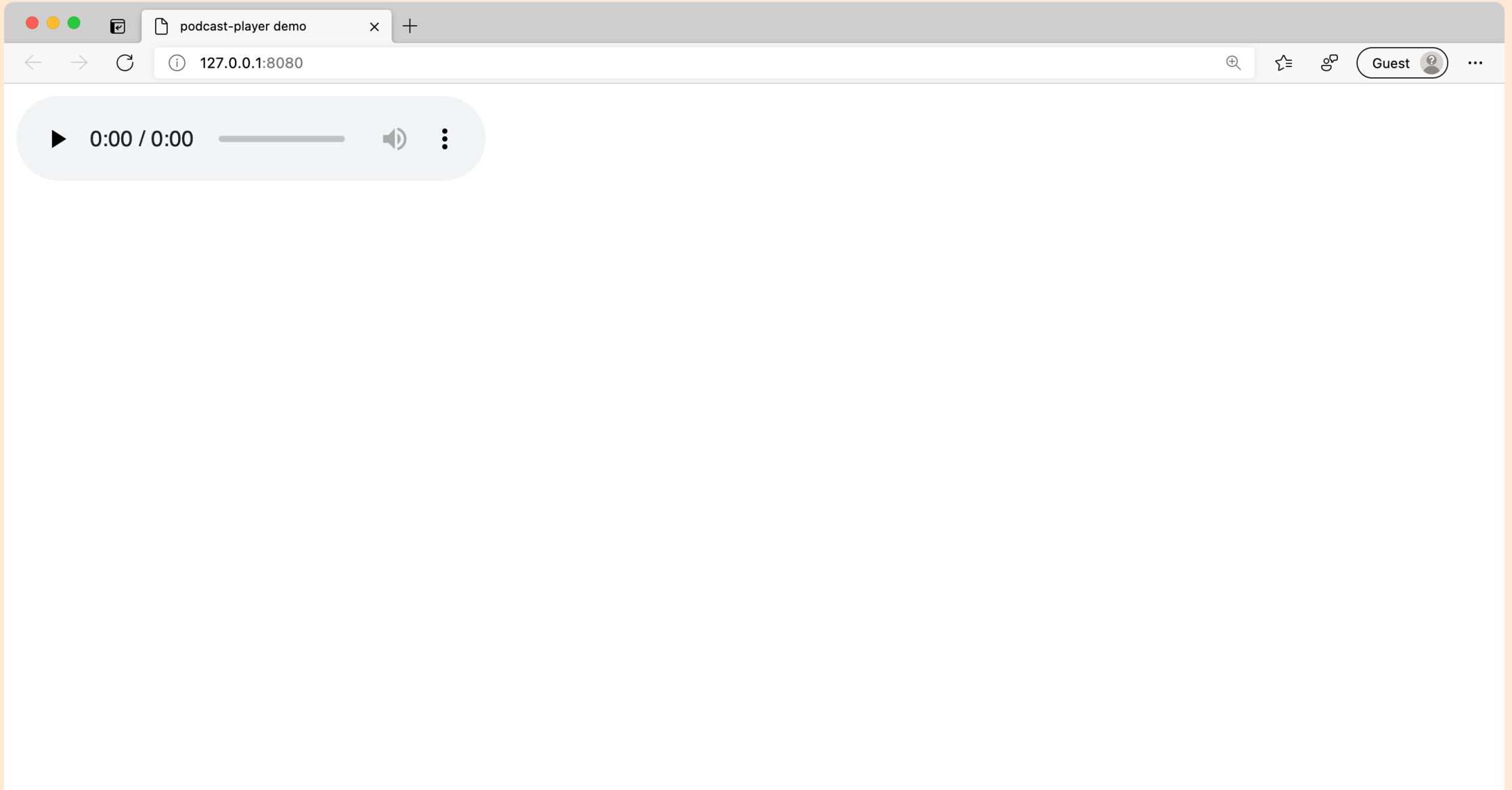
Web Components are great for Progressive Enhancement



I have strong opinions about podcast players



The native <audio> player

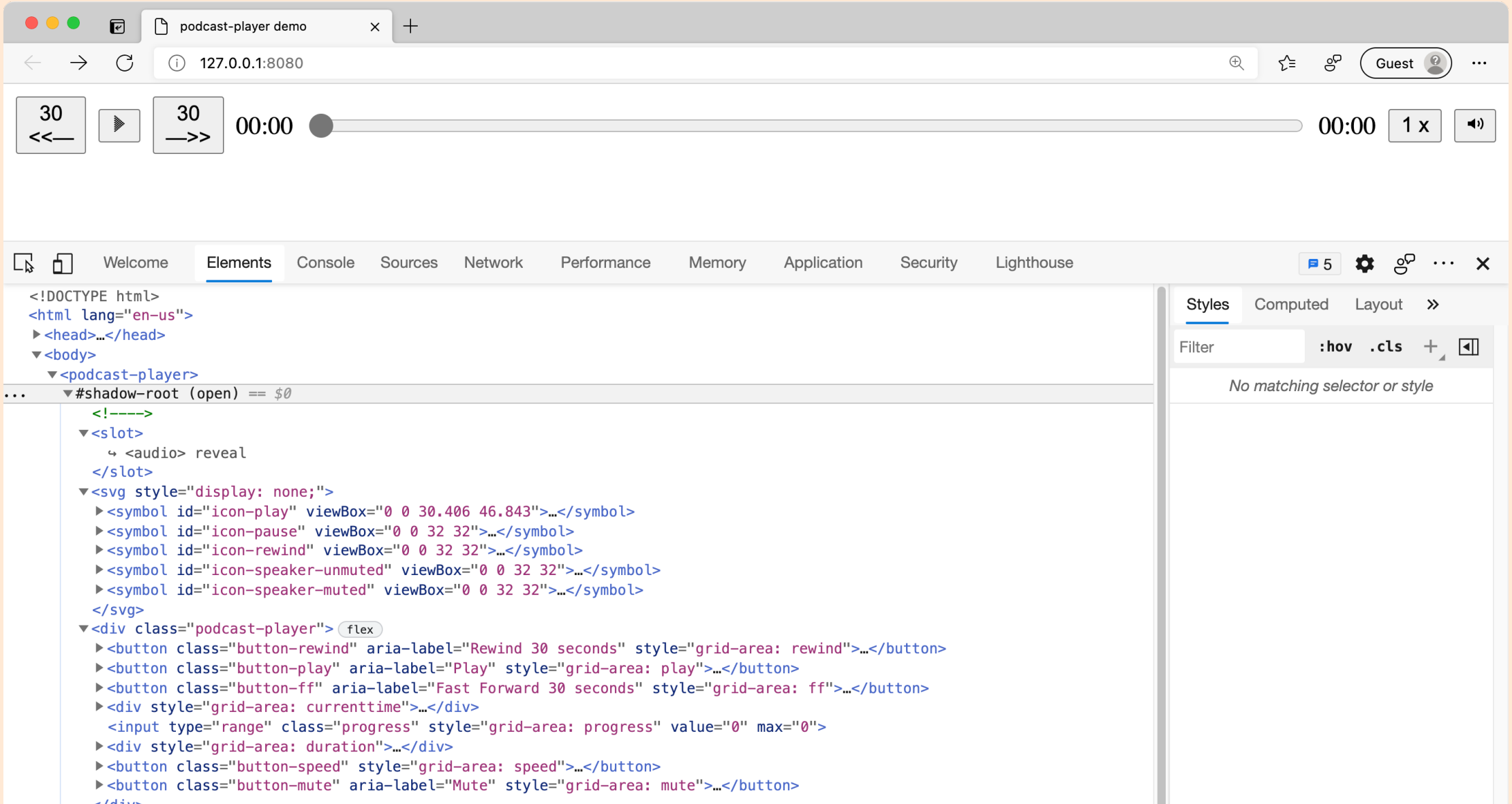


<podcast-player>

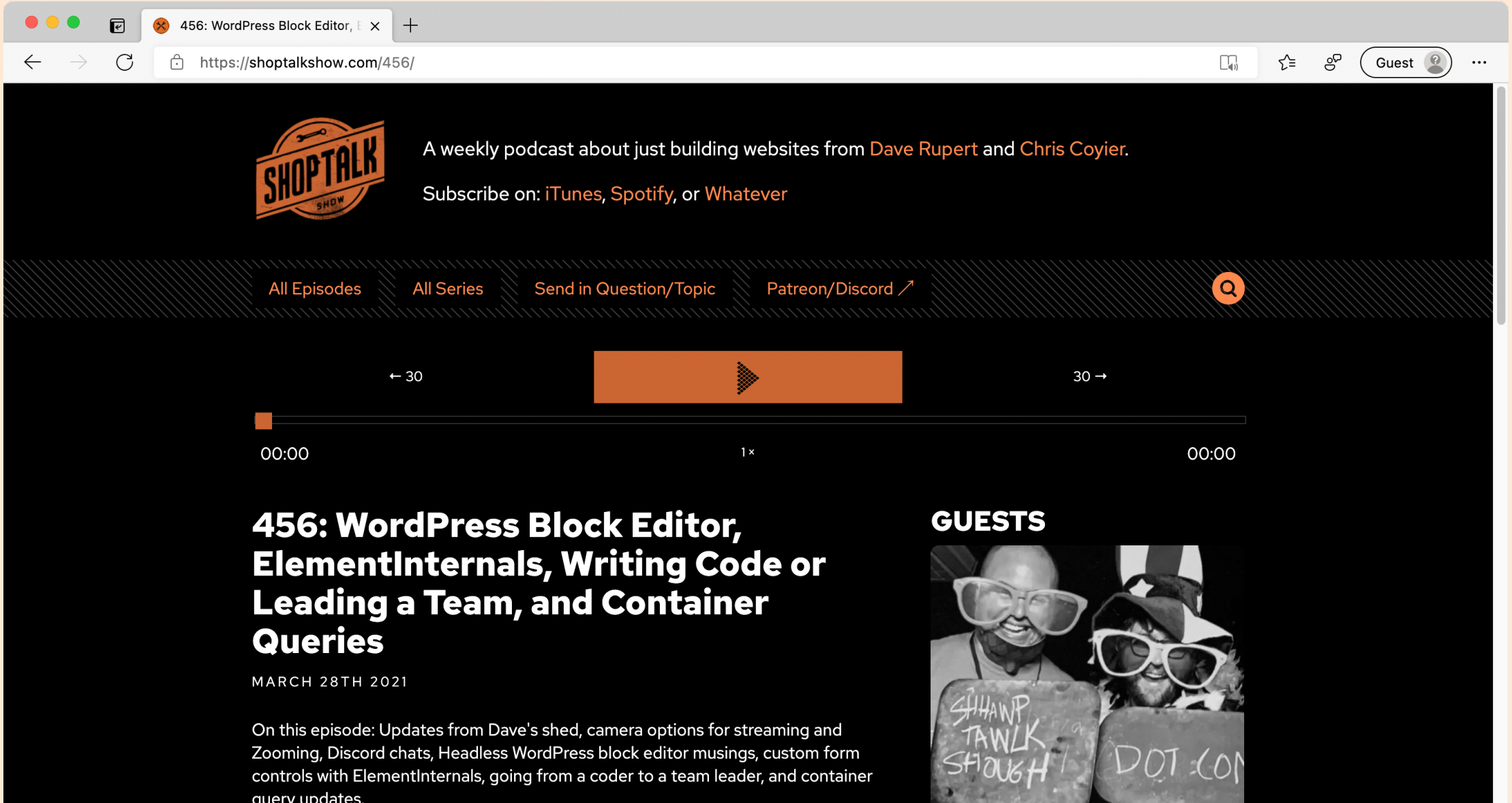
```
<podcast-player>
    <audio src="path/to/my.mp3" controls></audio>
</podcast-player>

<script type="module" src="path/to/podcast-player.js"></script>
```

<podcast-player>

<podcast-player>

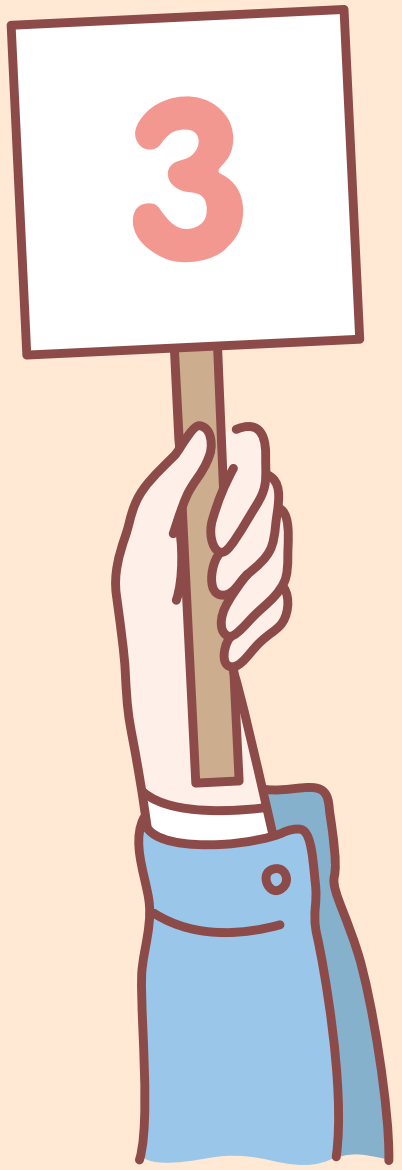


<podcast-player>



Assertion #3

Web Components will let us make simpler, more accessible, and more maintainable websites.



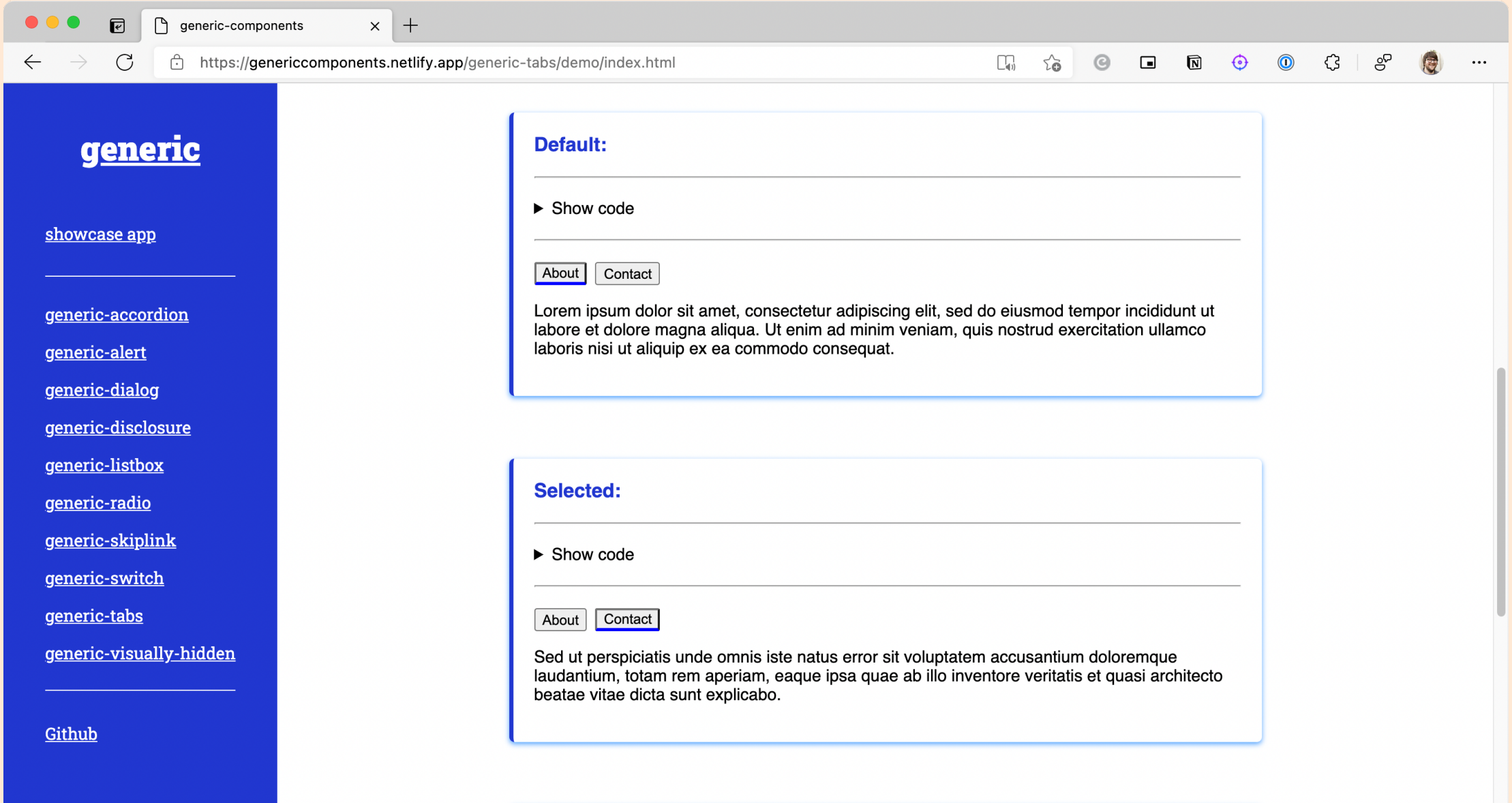
Wait. Wait. Hold up. Wait.

Are you saying Web Components would allow us to create and share **accessible prebuilt components** *without the need for specific JavaScript libraries* and *overly complicated build processes*?!

Yes. I am.

That’s the vision.

Generic Components



<generic-tabs>

```
<generic-tabs>
  <button slot="tab">About</button>
  <button slot="tab">Contact</button>

  <div slot="panel">
    <p>Lorem ipsum dolor sit amet….</p>
  </div>
  <div slot="panel">
    <p>Sed ut perspiciatis unde omnis….</p>
  </div>
</generic-tabs>
```

<generic-tabs>

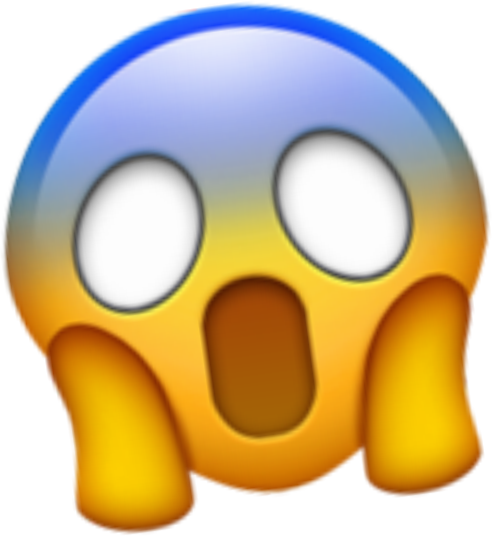
```
<generic-tabs>
  <h2>About</h2>
  <div>
    <p>Sed ut perspiciatis unde …</p>
  </div>

  <h2>Contact</h2>
  <div>
    <p>Ut enim ad minima veniam...</p>
  </div>
</generic-tabs>
```

[whatwg] Suggestion: Implementation of Tabbed Forms

This message: [Message body] [Respond] [More options]
Related messages: [Next message] [Previous message] [In reply to] [Next in thread] [Replies]

From: Ian Hickson <ian@hixie.ch>
Date: Tue, 17 Aug 2004 15:05:47 +0000 (UTC)
Message-ID: <Pine.LNX.4.61.0408171501020.16813@dhalsim.dreamhost.com>

```
On Tue, 17 Aug 2004, Matthew Raymond wrote:
>
> Ian Hickson wrote:
> > At the moment the spec says it would be:
> >
> >    <switch>
> >      <section>
> >        <h1>Exclusive Section 1</h1>
> >      </section>
> >      <section>
> >        <h1>Exclusive Section 2</h1>
> >      </section>
> >    </switch>
>
> I presume there is a mechanism for selecting a specific section. If so,
> would there be markup that would serve a function similar to a VB
> tabstrip? Will there be a markup solution to selecting a section?

The above is not a tabbed system, it's just a list of mutually exclusive
sections. Tabs are not mutually exclusive, they are just an unordered
group of related sections that are usually shown such that only one is
visible at any one time.

Anyway, the spec currently has a separate section for tabs, which says
that the tabbed version of the above would be:

    <tabbox>
      <section>
        <h1>Section 1</h1>
      </section>
      <section>
        <h1>Section 2</h1>
      </section>
    </tabbox>

The name <tabbox> will probably be changed to <group>. See the spec for
better examples. (Graceful degradation as your example had is possible
```



Assertion #4

Web Components are a gateway to a new dimension.

<model-viewer>

https://modelviewer.dev

Guest

<model-viewer>

Easily display interactive 3D models on the web & in AR

Quick Start

```
<!-- Import the component -->
<script type="module" src="https://unpkg.com/@google/model-viewer/dist/model-viewer.min.js"></script>

<!-- Use it like any other HTML element -->
<model-viewer src="shared-assets/models/Astronaut.glb" alt="A 3D model of an astronaut" auto-rotate camera-controls></model-viewer>
```

minzipped size 202.6 KB release v1.6.0

Follow @modelviewer 1.7k Star 2.7k

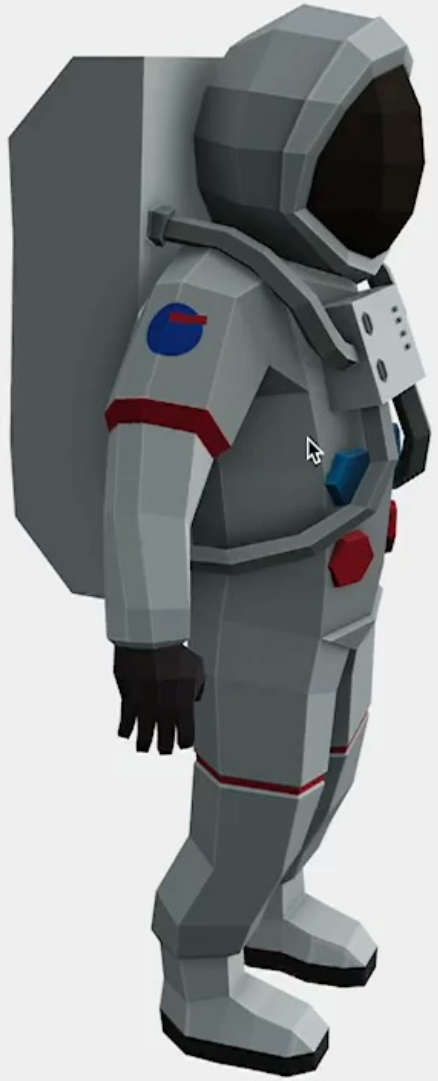
Documentation

API documentation & examples

Model Editor

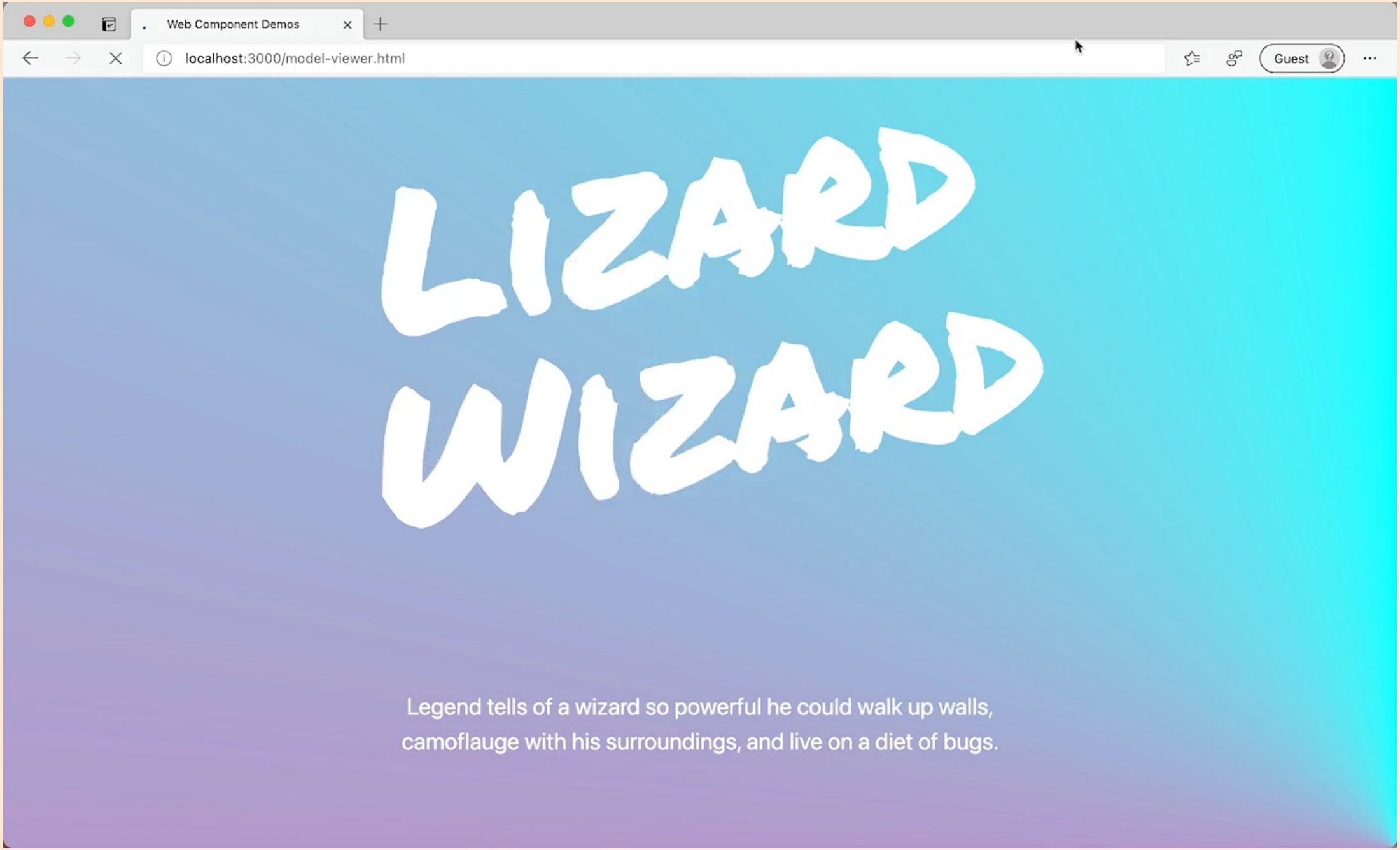
Use the Model Editor to prepare your GLB models for the web

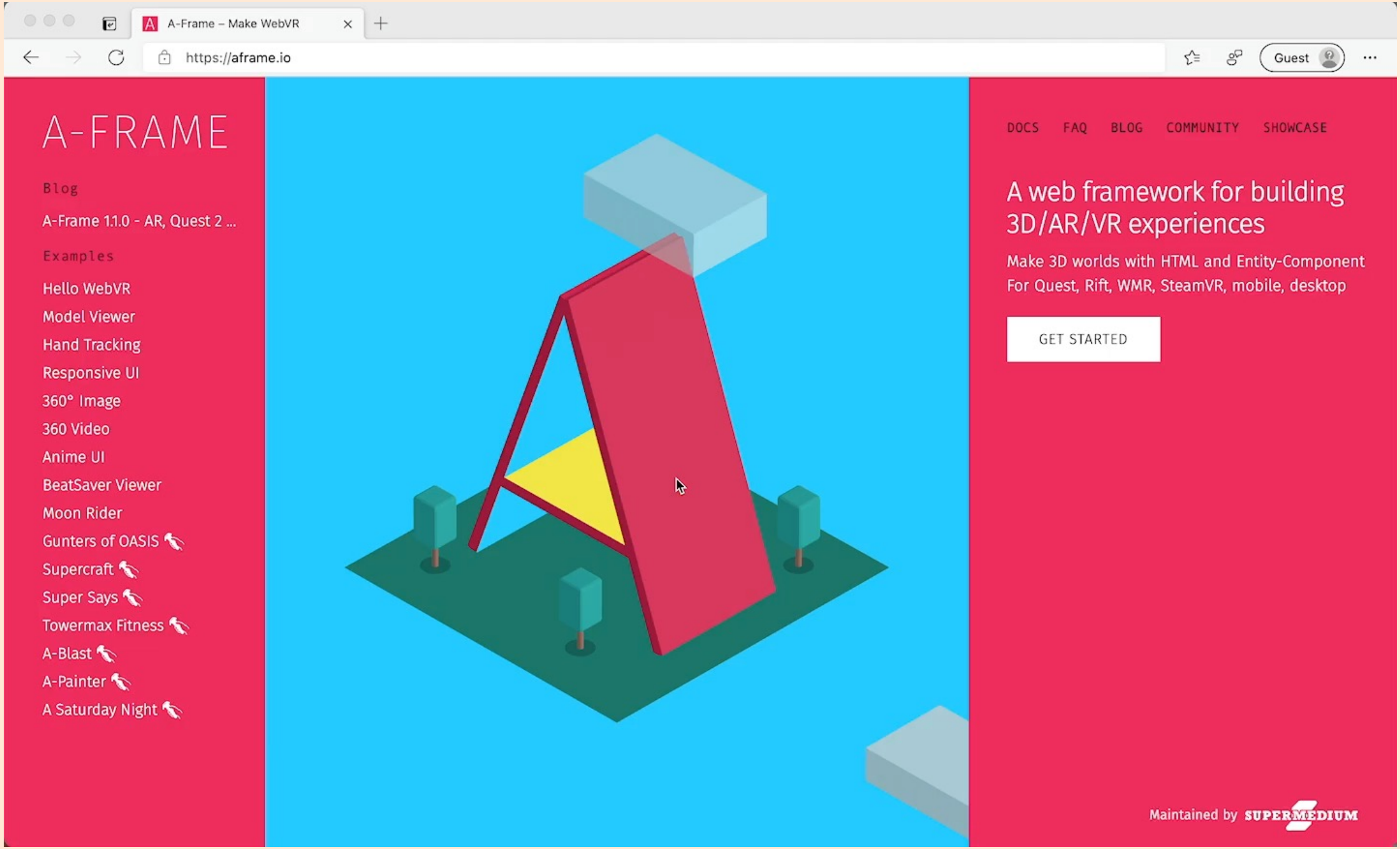
Interactive Viewer (Legacy)



<model-viewer>

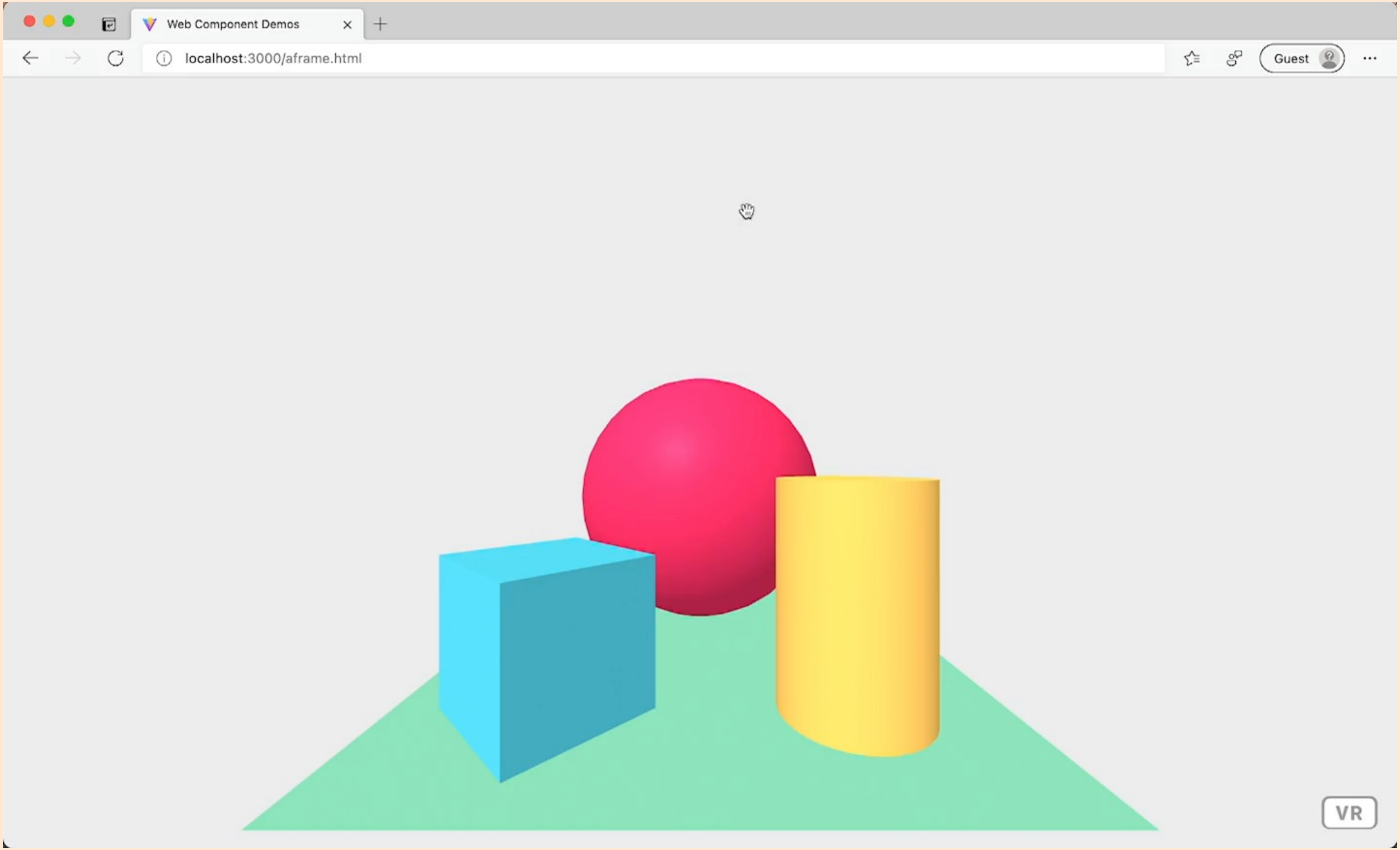
```
<script type="module" src="path/to/model-viewer.min.js"></script>

<model-viewer src="path/to/Astronaut.glb"
              alt="A 3D model of an astronaut"
              auto-rotate
              camera-controls>
</model-viewer>
```

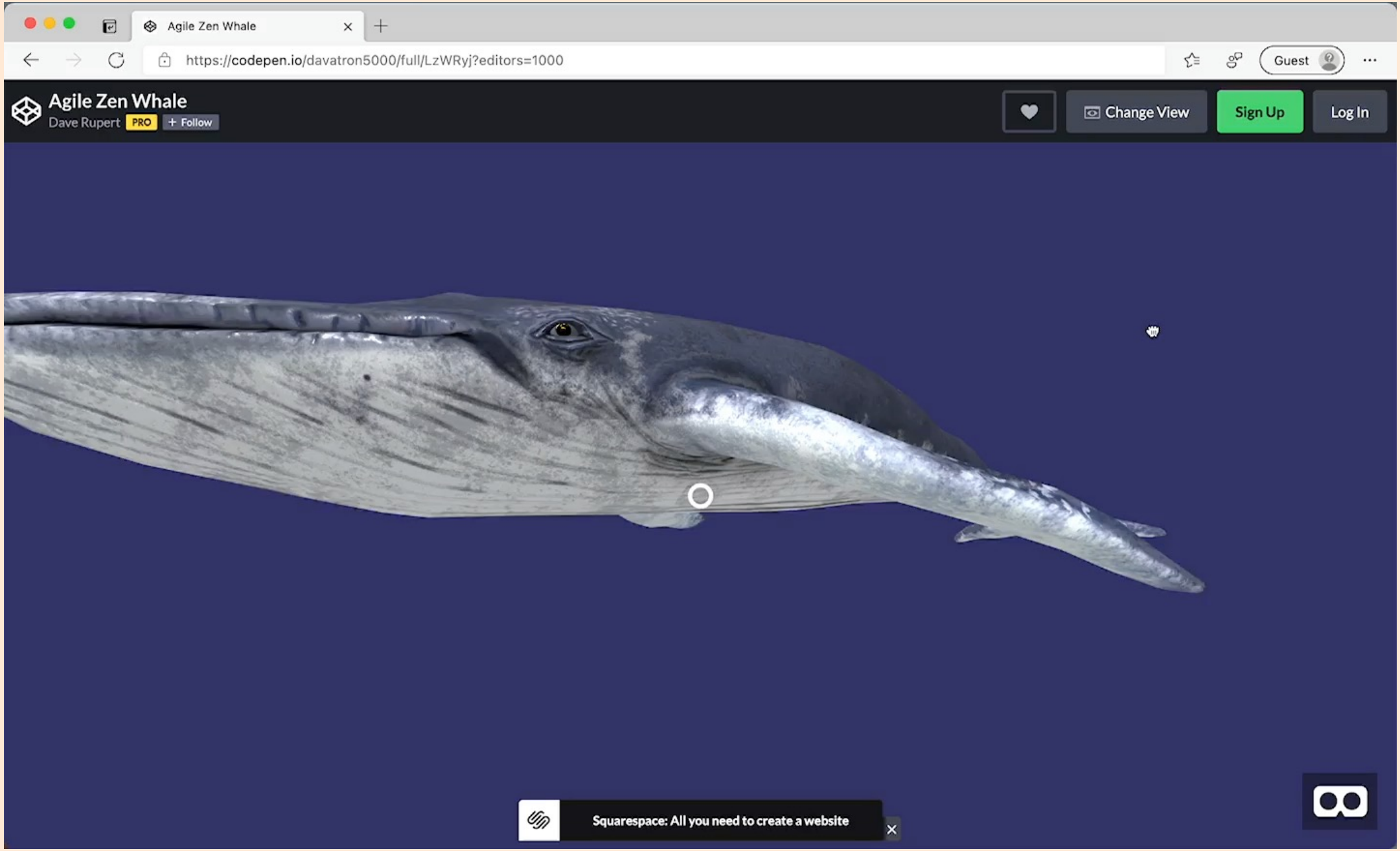

Web Component Demos
localhost:3000/model-viewer.html
Guest
LIZARD WIZARD
Legend tells of a wizard so powerful he could walk up walls,
camoflauge with his surroundings, and live on a diet of bugs.


A-Frame – Make WebVR
https://aframe.io
Guest
A-FRAME
Blog
A-Frame 1.1.0 - AR, Quest 2 ...
Examples
Hello WebVR
Model Viewer
Hand Tracking
Responsive UI
360° Image
360 Video
Anime UI
BeatSaver Viewer
Moon Rider
Gunters of OASIS
Supercraft
Super Says
Towermax Fitness
A-Blast
A-Painter
A Saturday Night
DOCS FAQ BLOG COMMUNITY SHOWCASE
A web framework for building 3D/AR/VR experiences
Make 3D worlds with HTML and Entity-Component
For Quest, Rift, WMR, SteamVR, mobile, desktop
GET STARTED
Maintained by SUPERMEDIUM

<a-frame>

```
<script src="path/to/aframe.min.js"></script>

<a-scene>
    <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
    <a-box position="-1 0.5 -3" rotation="0 45 0" width="1" height="1" depth="1" color="#4CC3D9"></a-box>
    <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
    <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
    <a-sky color="#ECECEC"></a-sky>
</a-scene>
```

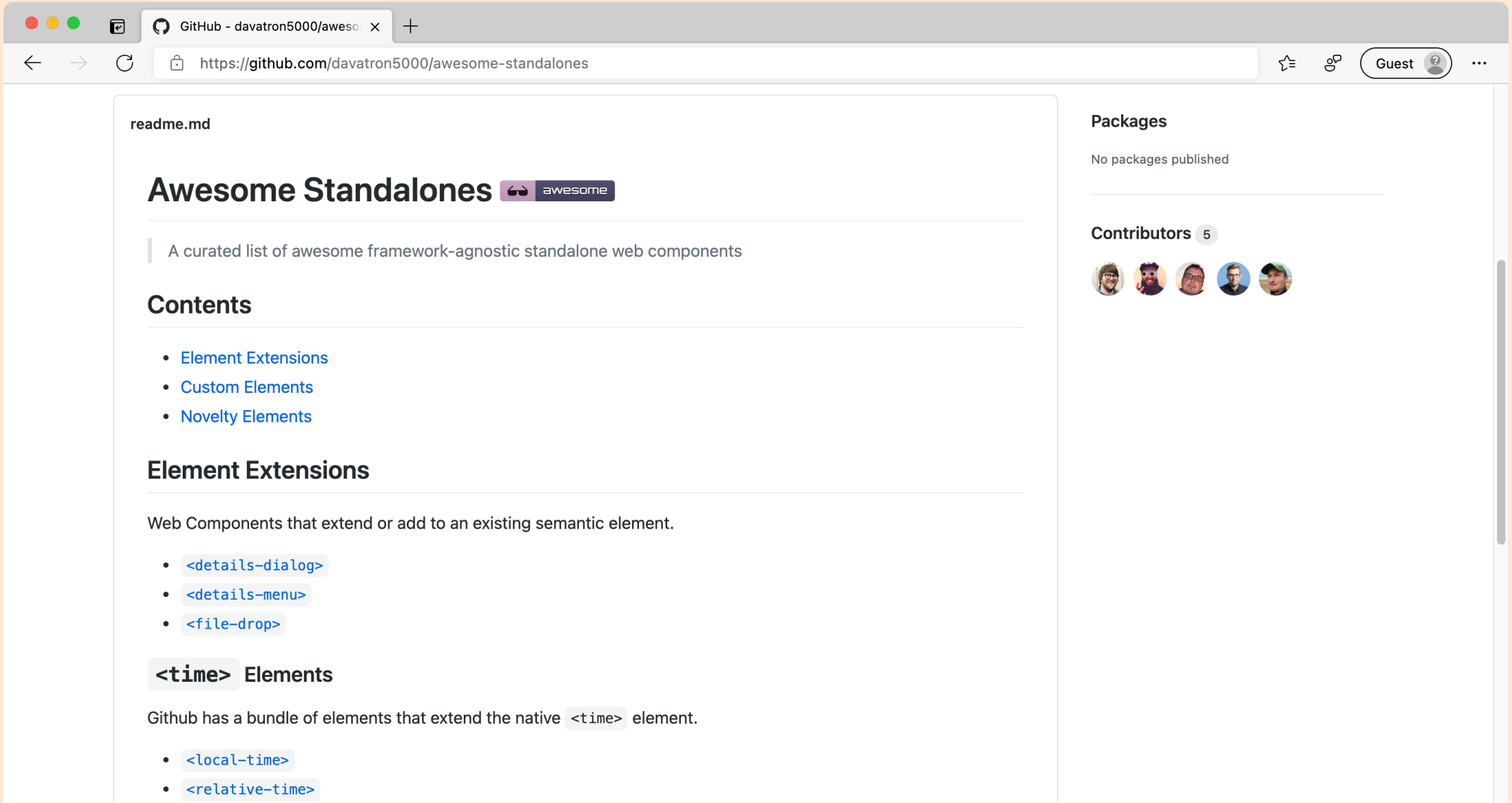

Web Component Demos
localhost:3000/aframe.html
Guest
VR


Agile Zen Whale
https://codepen.io/davatron5000/full/LzWRyj?editors=1000
Guest
Agile Zen Whale
Dave Rupert
PRO
+ Follow
Change View
Sign Up
Log In
Squarespace: All you need to create a website

How do you find cool Web Components?

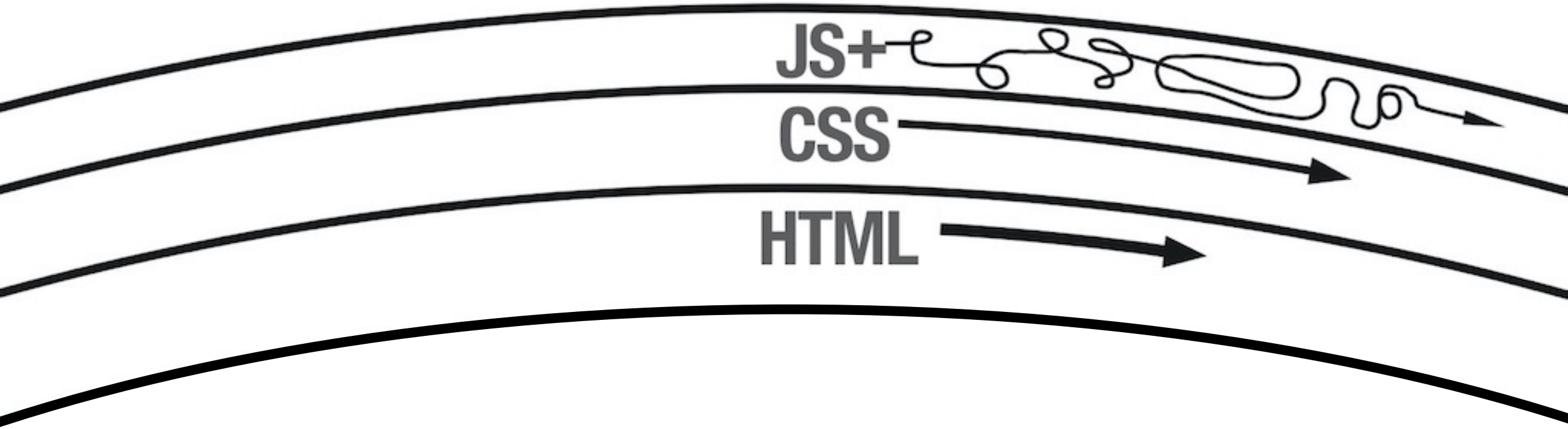
Me, personally? Poor work/life balance. You?

🕶 Awesome Standalones

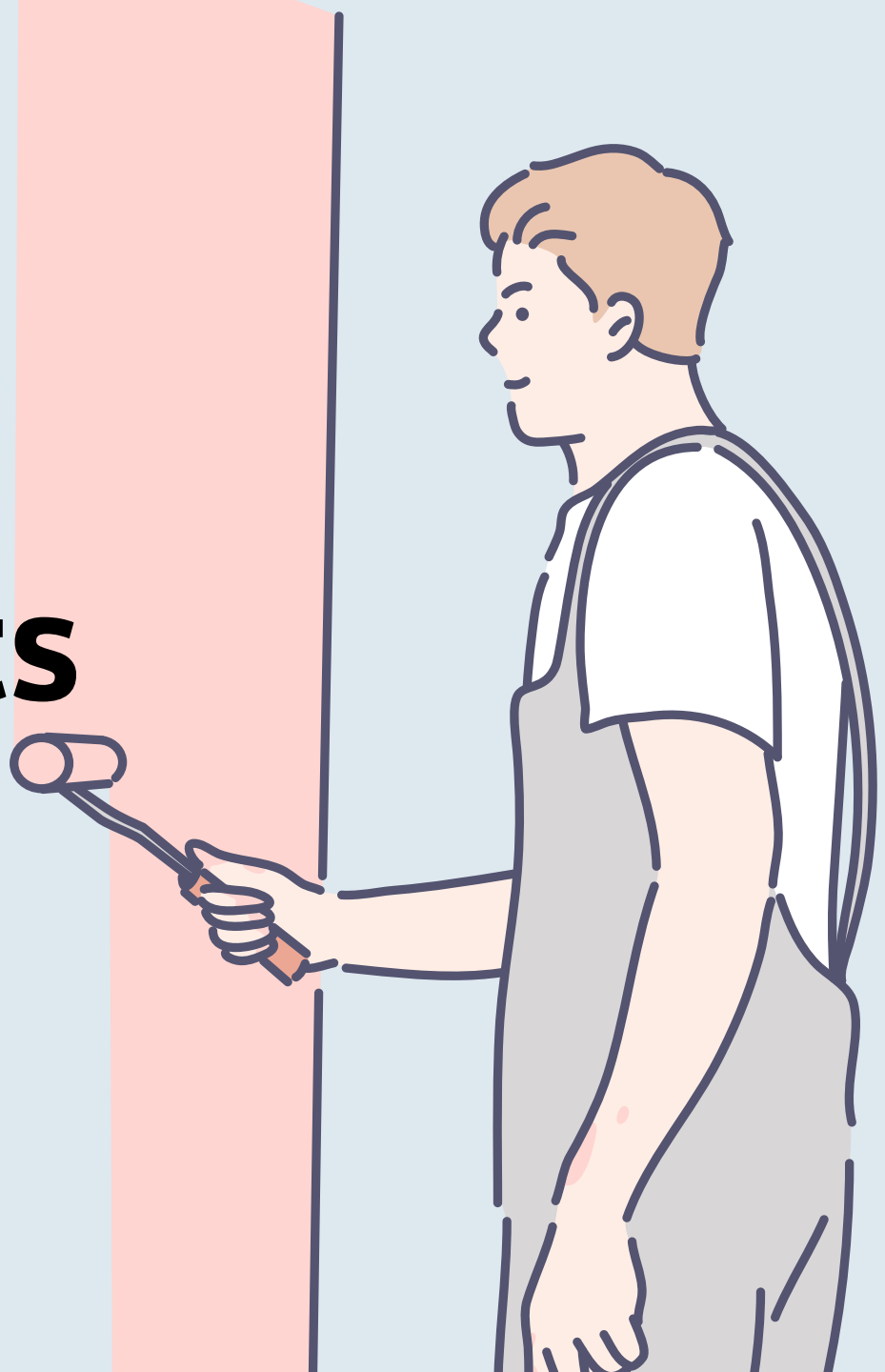


Summary of what's good about Web Components

- ✓ Easy as HTML
- ✓ Great reusability
- ✓ Better experiences
- ✓ Progressively enhanced
- ✓ No build tools
- ✓ "Encapsulated"


JS+
CSS
HTML

Styling Web Components with CSS



Truth be told...

This part isn’t very fun.

What pierces the Shadow DOM?

```
<custom-alert>
  <p>Light DOM goes here</p>
</custom-alert>
```

What pierces the Shadow DOM?

```
<custom-alert>
  <p>Light DOM goes here</p>
    <img src="path/to/icon.svg"/>
    <slot></slot>
    <button>Close</button>
</custom-alert>
```

What pierces the Shadow DOM?

```
body p {
  transform: rotate(-10deg);
}

custom-alert button {
  background: pink;
}

button {
  color: pink;
}

custom-alert {
  --bg: red;
  --text: white;
}
```

What pierces the Shadow DOM?

```
custom-alert button {
  background: pink;
}
```

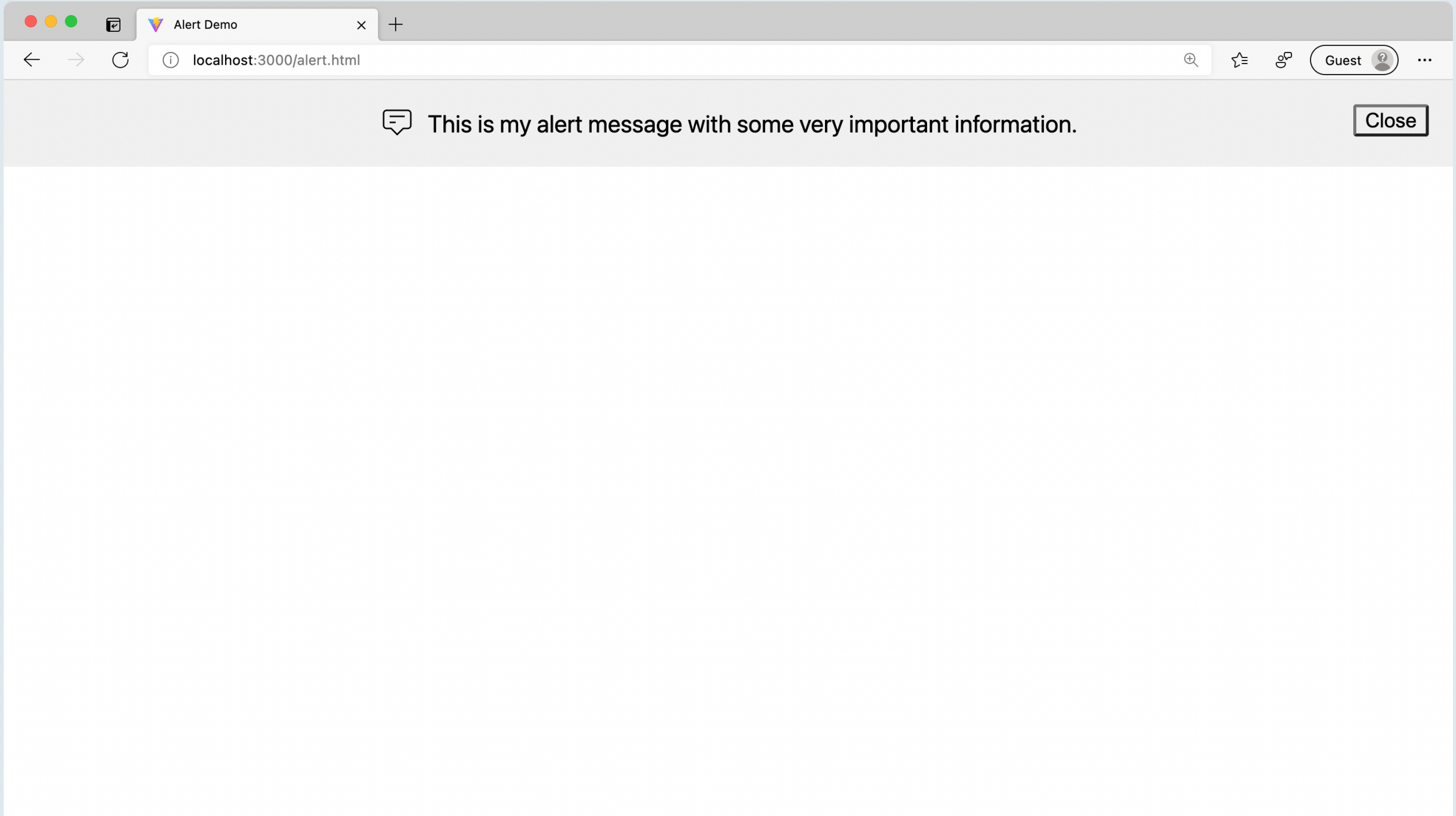

```
body p {
  transform: rotate(-10deg);
}
```

```
button {
  color: pink;
}
```

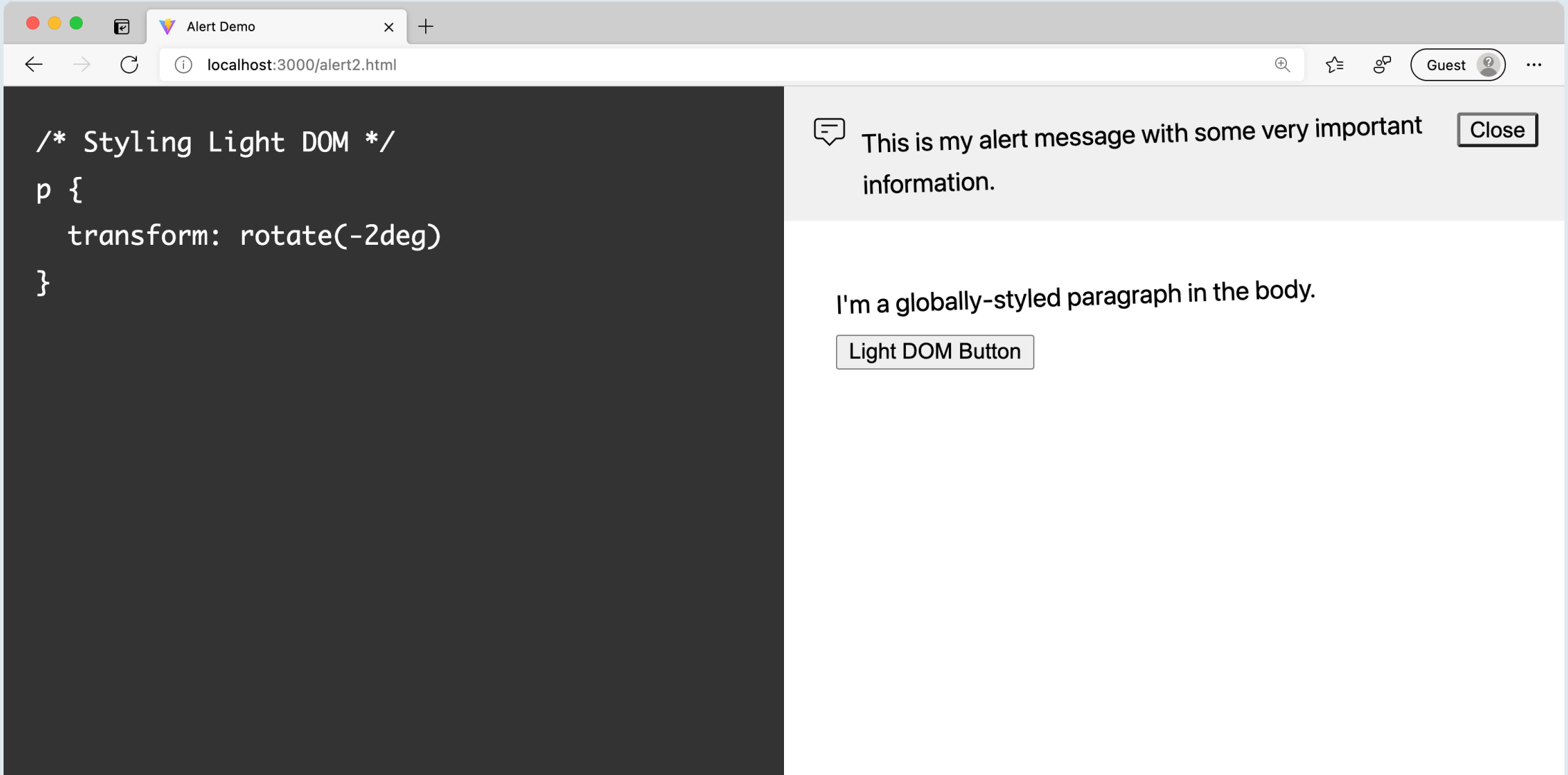
```
custom-alert {
  --bg: red;
  --text: white;
}
```

A loss of control

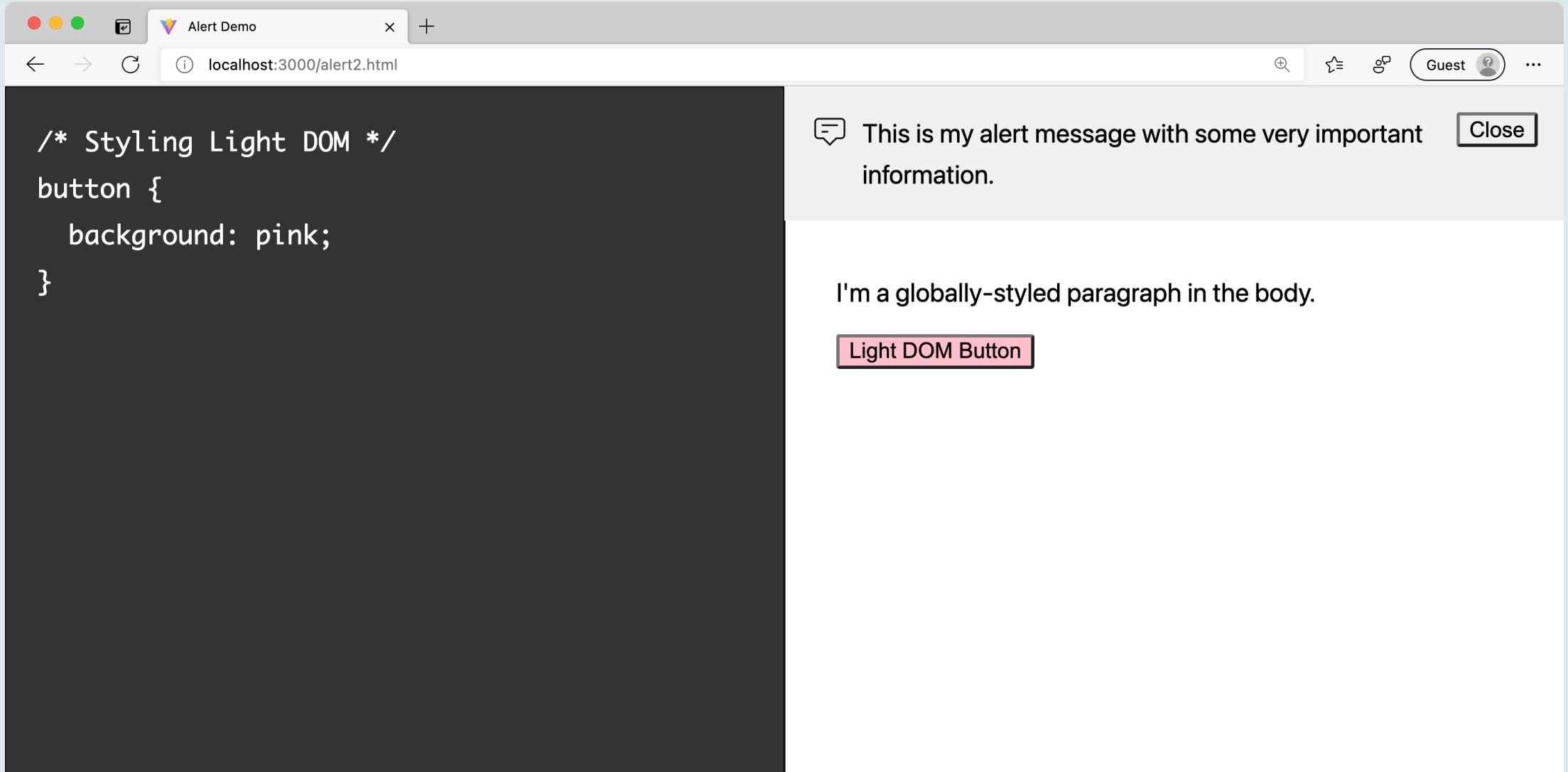
When you’re comfortable with the global, cascading nature of CSS it can be quite frustration when it quits working.


Alert Demo
localhost:3000/alert.html
Guest
This is my alert message with some very important information.
Close

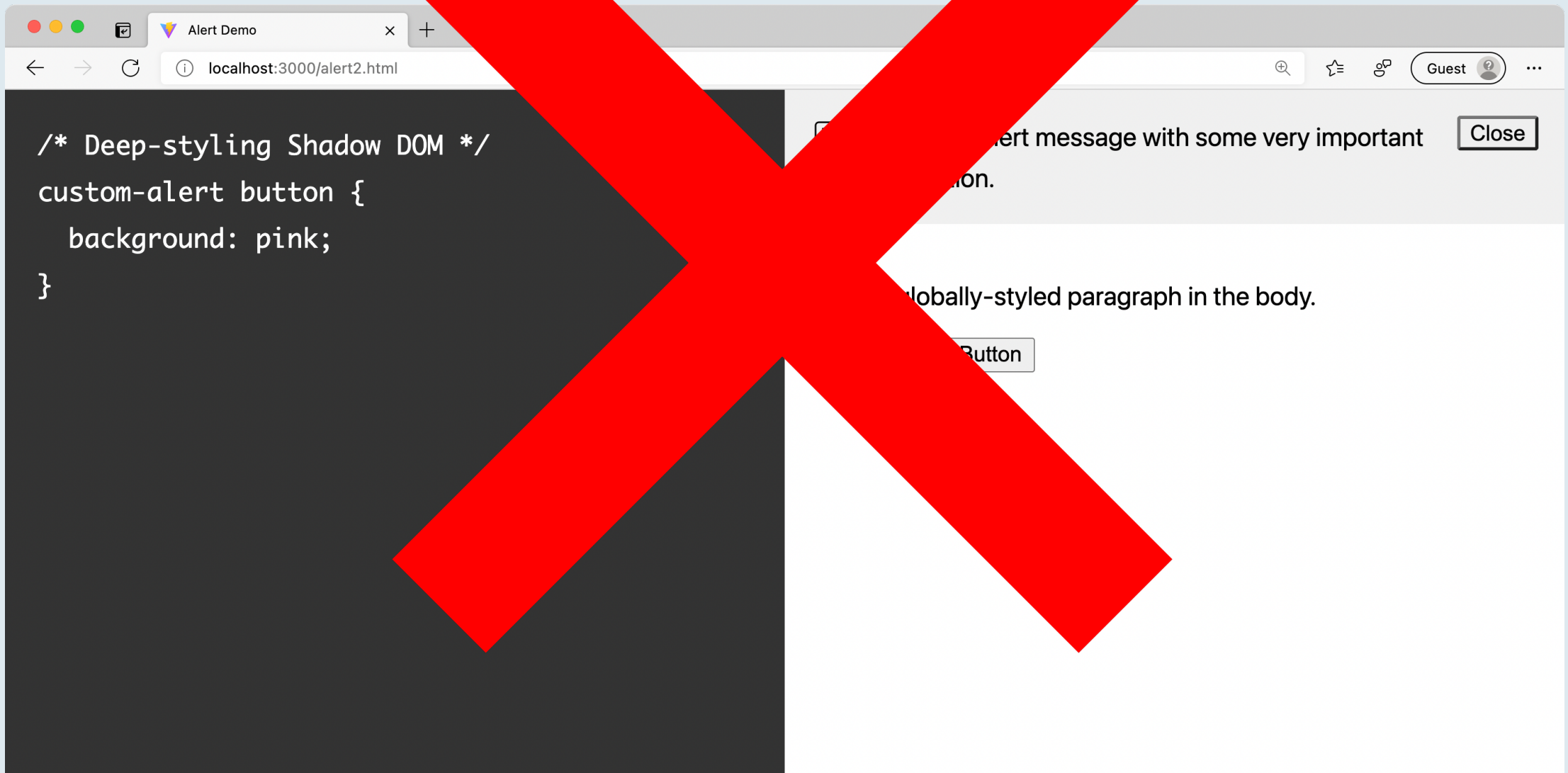
Styling Light DOM?



What about that <button>?



Deep Styling Shadow Elements?



But if this wasn’t difficult enough…

Some styles DO bleed through the shadow boundary!

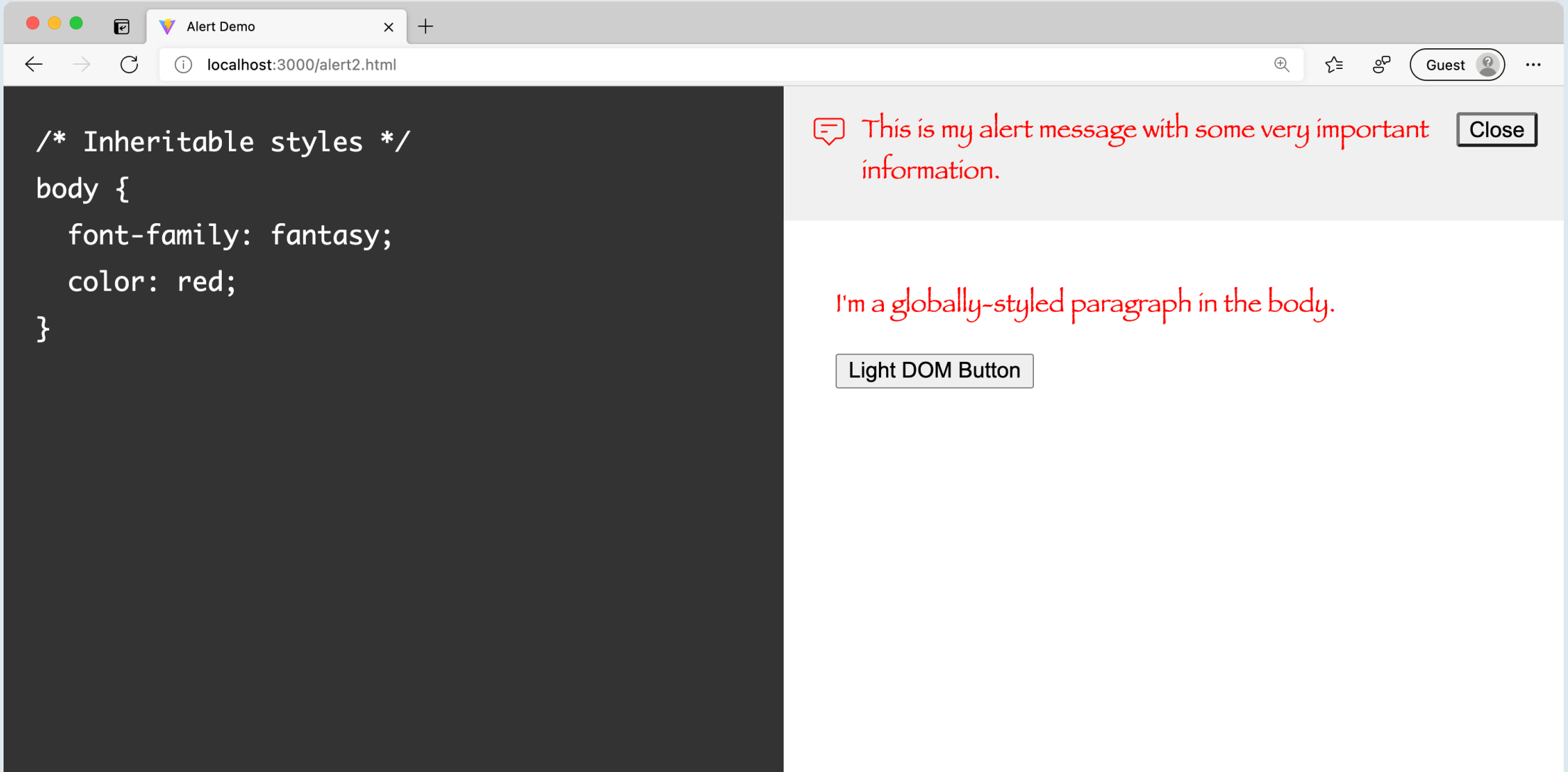
Inheritable Styles

- border-collapse
- border-spacing
- caption-side
- color
- cursor
- direction
- empty-cells
- font-family
- font-size
- font-style
- font-variant
- font-weight
- font-size-adjust
- font-stretch
- font
- letter-spacing
- line-height
- list-style-image
- list-style-position
- list-style-type
- list-style
- orphans
- quotes
- tab-size
- text-align
- text-align-last
- text-decoration-color
- text-indent
- text-justify
- text-shadow
- text-transform
- visibility
- white-space
- widows
- word-break
- word-spacing
- word-wrap

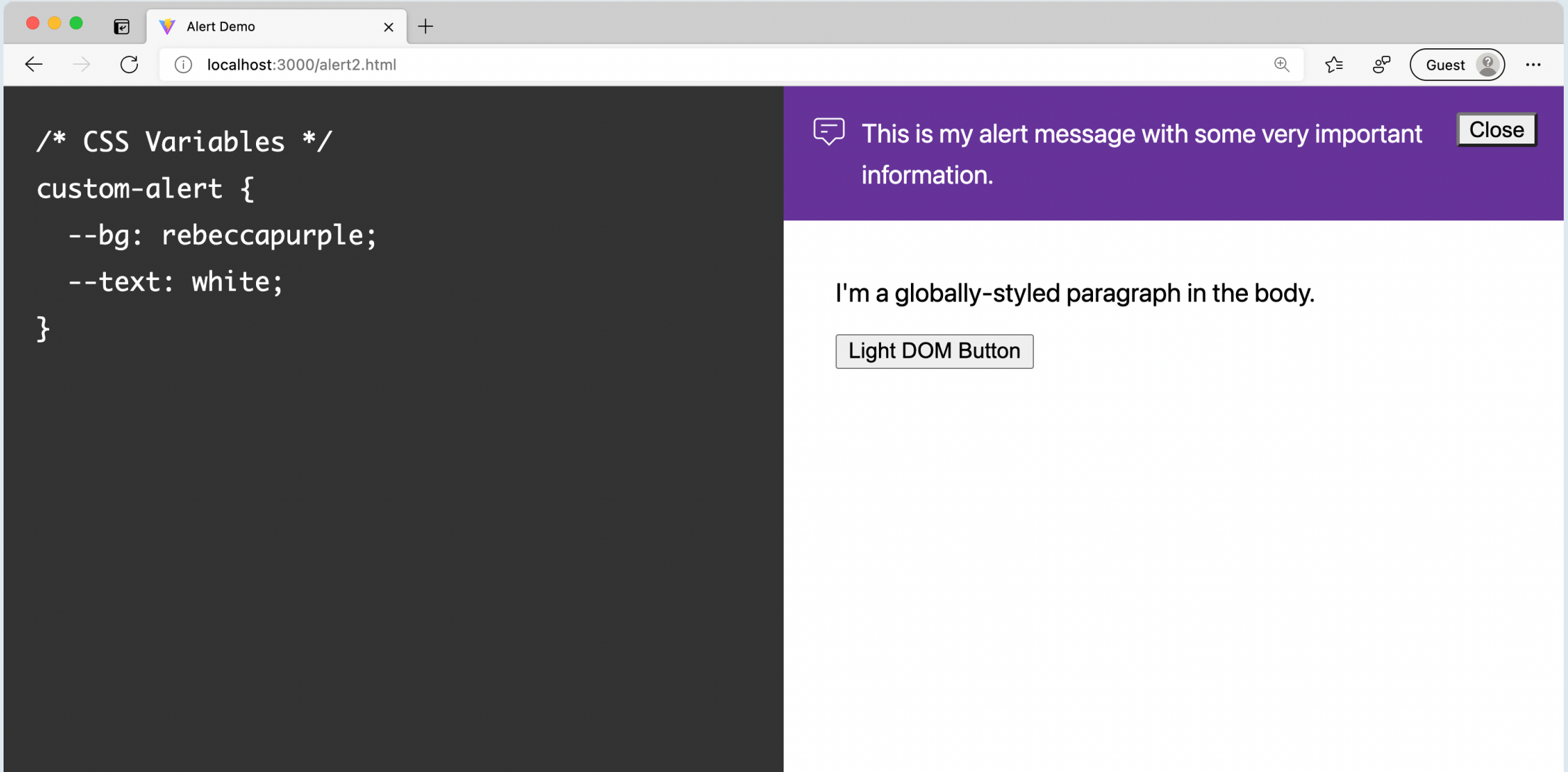
Inheritable Styles

- border-collapse
- border-spacing
- caption-side
- **color**
- **cursor**
- direction
- empty-cells
- **font-family**
- **font-size**
- **font-style**
- **font-variant**
- **font-weight**
- font-size-adjust
- font-stretch
- **font**
- **letter-spacing**
- **line-height**
- list-style-image
- list-style-position
- list-style-type
- list-style
- orphans
- quotes
- tab-size
- **text-align**
- text-align-last
- text-decoration-color
- text-indent
- text-justify
- text-shadow
- **text-transform**
- visibility
- white-space
- widows
- word-break
- word-spacing
- word-wrap

Inheritable Styles?



CSS Variables?



<two-up> styling API

```
/* Two Up Styling API */
two-up {
  --accent-color: #777;
  --track-color: #777;
  --thumb-color: #777;
  --thumb-background: #fff;
  --thumb-size: 62px;
  --bar-size: 6px;
  --bar-touch-size: 30px;
}
```

CSS Variables are a great way to add a styling API

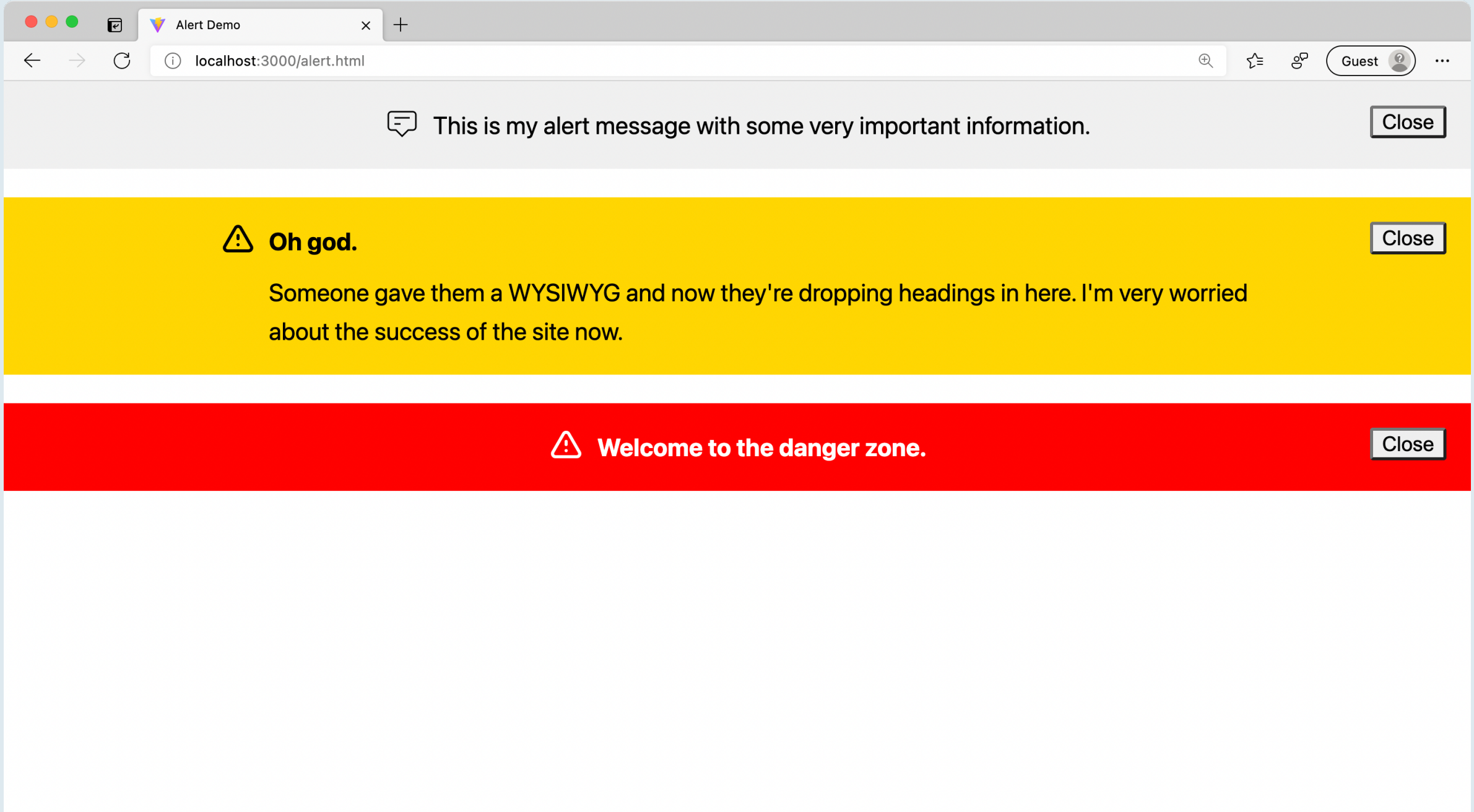
They offer the right amount of flexibility for most minor customizations.

More ways to offer styling APIs

A few “hooks” to let their components be customized.

Custom themes or classes

```
<custom-alert theme="warn">
  <p>Light DOM goes here</p>
</custom-alert>
```


Alert Demo
localhost:3000/alert.html
Guest
This is my alert message with some very important information.
Close
Oh god.
Someone gave them a WYSIWYG and now they're dropping headings in here. I'm very worried about the success of the site now.
Close
Welcome to the danger zone.
Close

I’m still kinda mad

About that button.

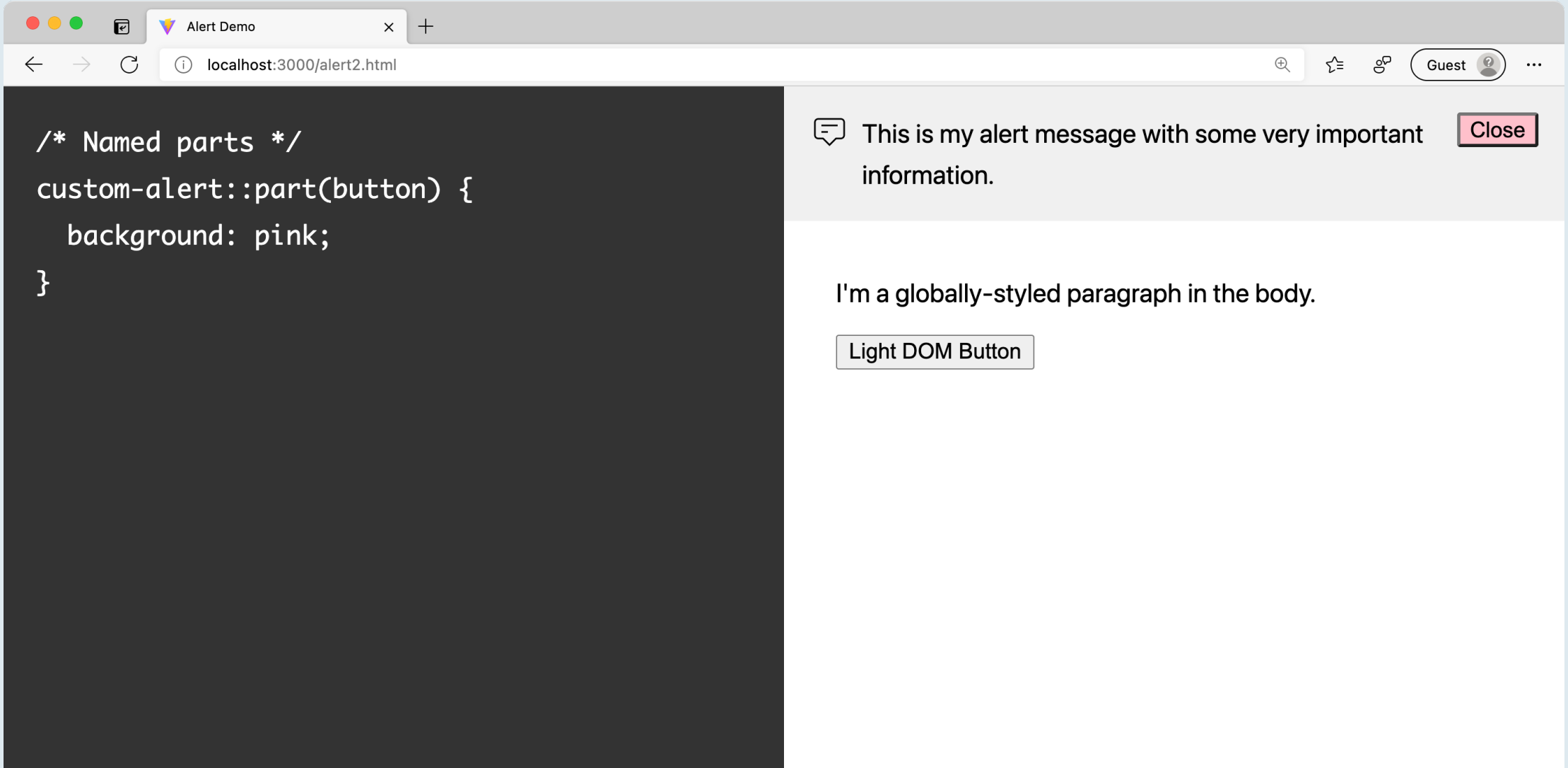
Named Shadow Parts

```
<custom-alert>
  <p>Light DOM goes here</p>
</custom-alert>
```

Named Shadow Parts

```
<custom-alert>
  <p>Light DOM goes here</p>
  <img name="icon" src="icon.svg"/>
  <slot></slot>
  <button name="button">Close</button>
</custom-alert>
```

Named Shadow Parts



Crossing the Shadow Boundary

Styling Light DOM

Deep Styling Shadow DOM

Inheritable Styles

CSS Variables

Theming

Named Shadow Parts

Crossing the Shadow Boundary

Deep Styling Shadow DOM

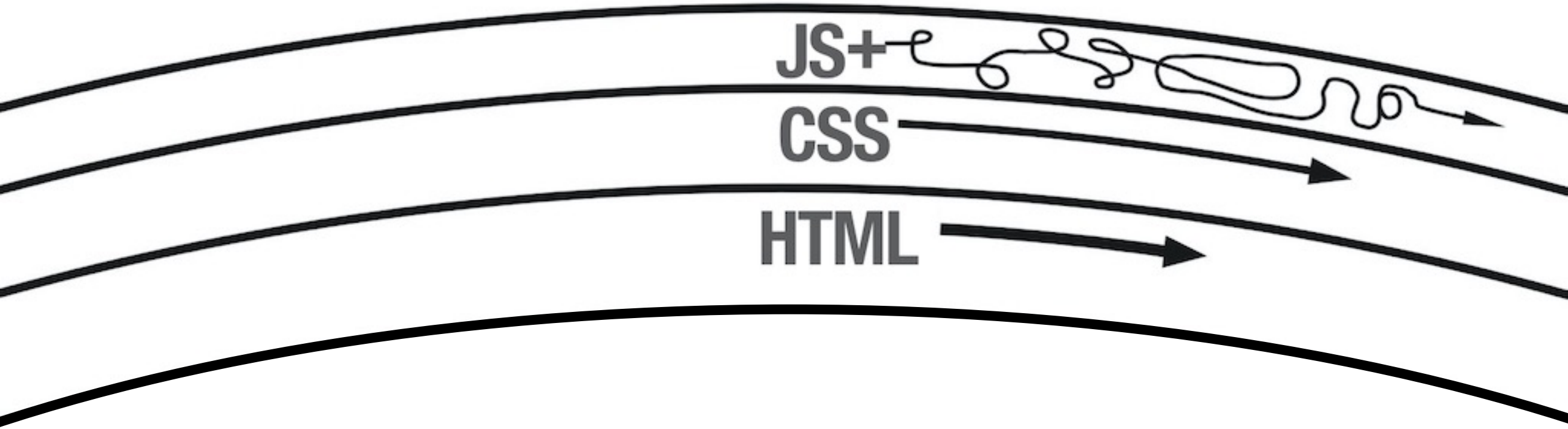
Styling Light DOM

Inheritable Styles

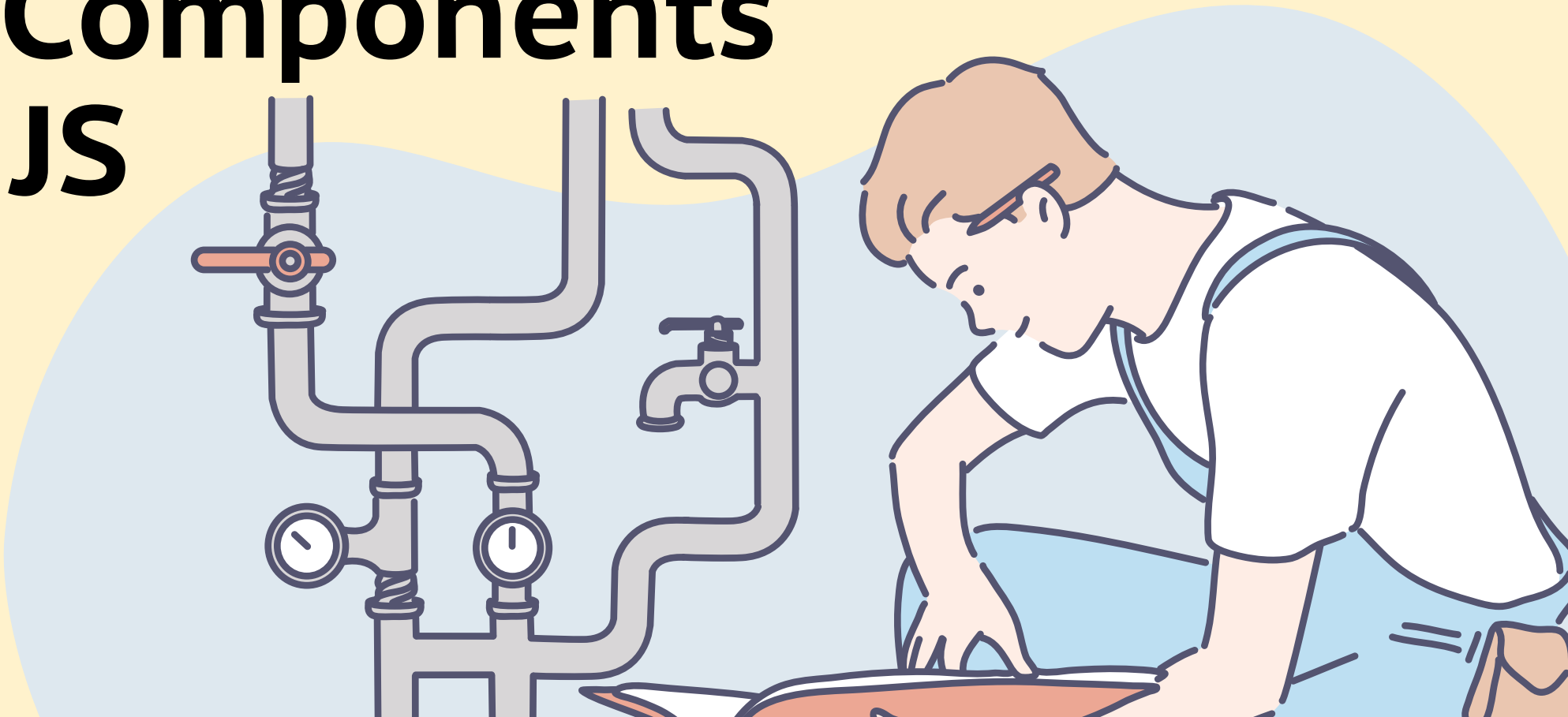
CSS Variables

Theming

Named Shadow Parts


JS+
CSS
HTML

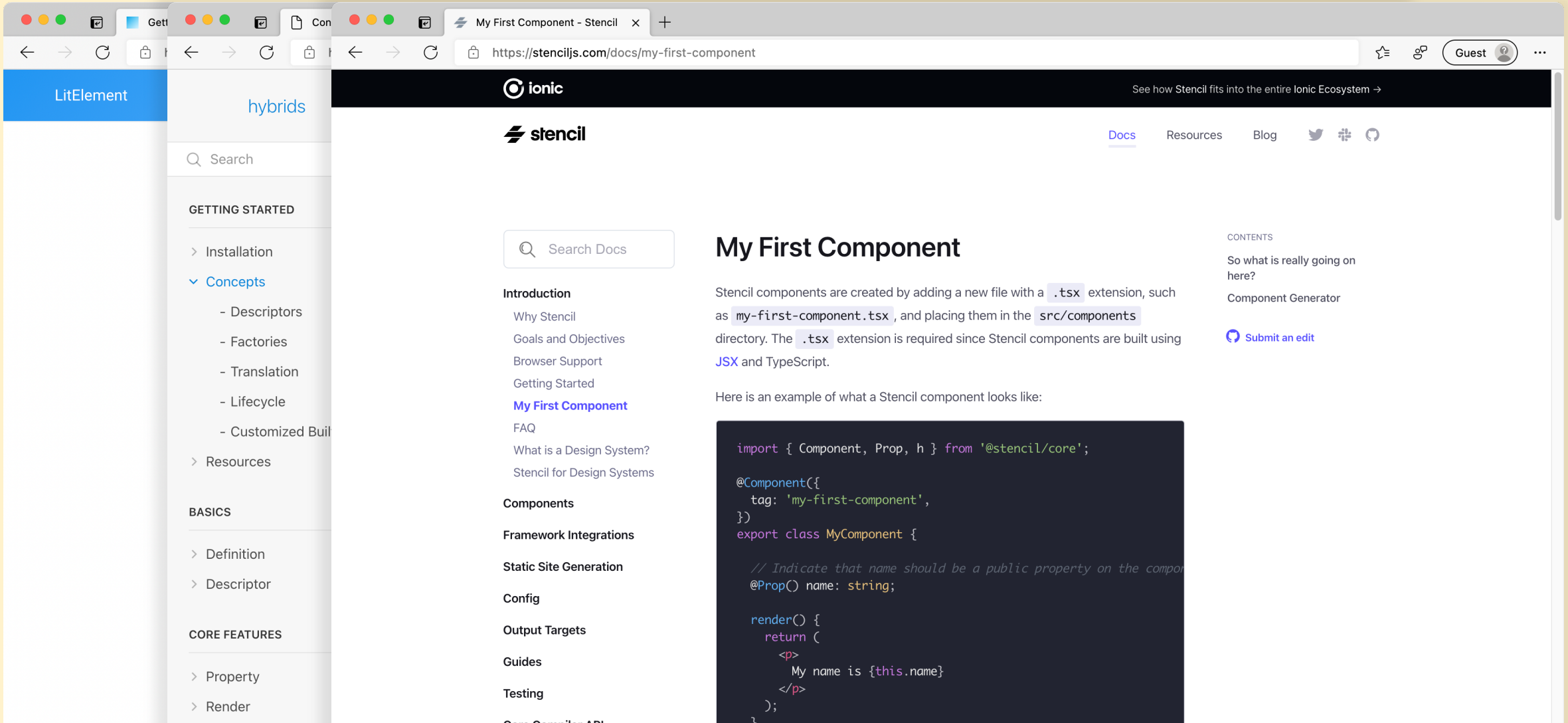
Writing Web Components with JS



Web Component Lifecycle

```
class CustomAlert extends HTMLElement {
    static get observedAttributes() {
        return ['icon', 'theme'];
    }

    constructor() {
        super();
    }

    connectedCallback()     {}      // componentDidMount
    disconnectedCallback() {}       // componentWillUnmount
    attributeChangedCallback(name, oldValue, newValue) {}
}

customElements.define('custom-alert', CustomAlert);
```

Small libraries (5~10kb) exist to make writing Web Components nicer.



A basic example in lit 🔥

```
<custom-alert icon="note" theme="warn">
    <p>Alert message goes here.</p>
</custom-alert>
```

A basic example

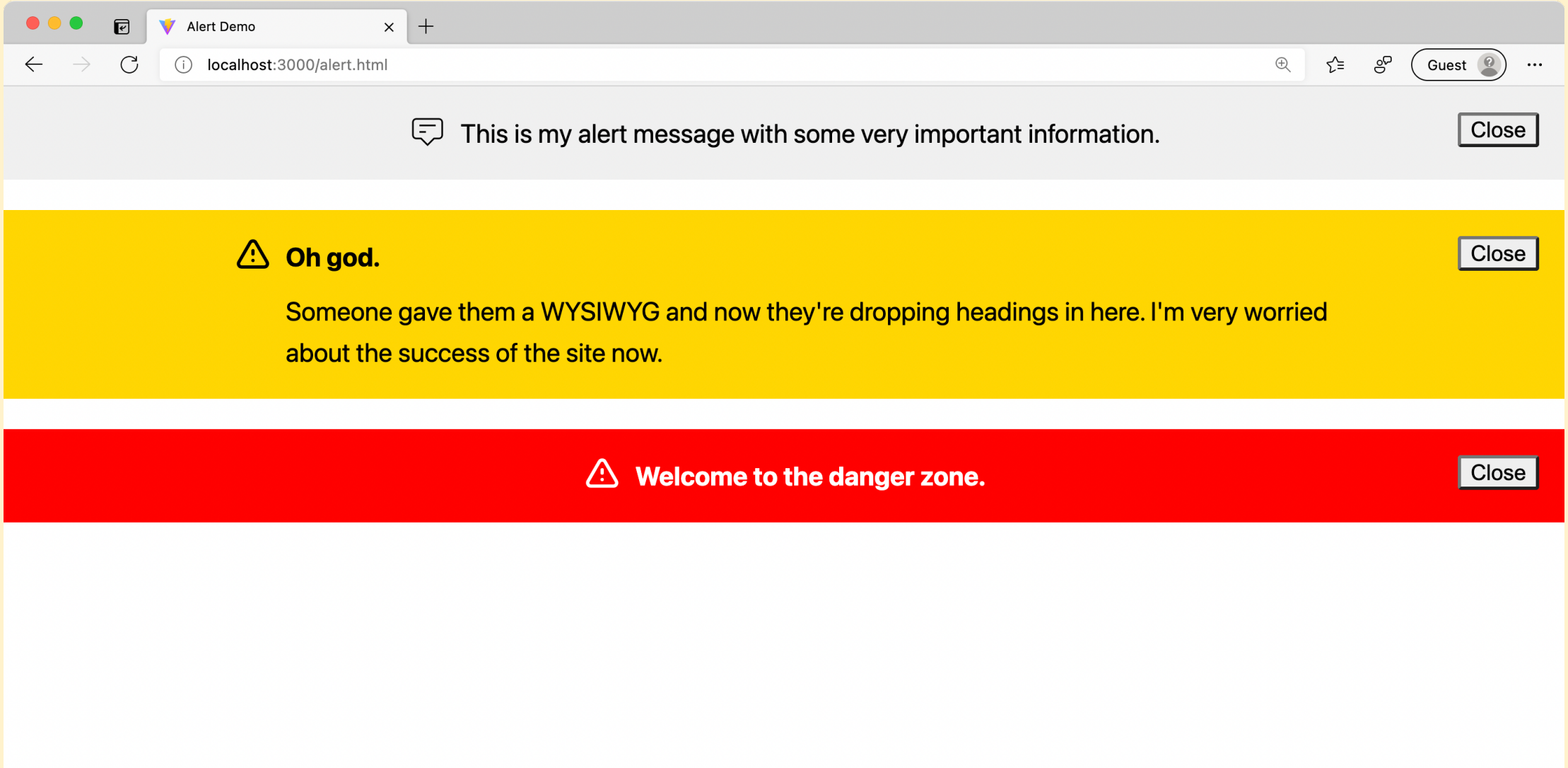
```
import { LitElement, html, css } from 'lit-element’

export class CustomAlert extends LitElement {
    static get styles() {
        return css`
            :host {
                background: var(--bg, #f0f0f0);
                color: var(--text);
                display: grid;
                gap: 0.5rem;
                grid-template-columns: 1fr 24px auto 1fr;
                align-items: start;
                padding: 1rem 0.5rem;
            }
            :host([hidden]) {
                display: none;
            }
            :host([theme="danger"]) {
```

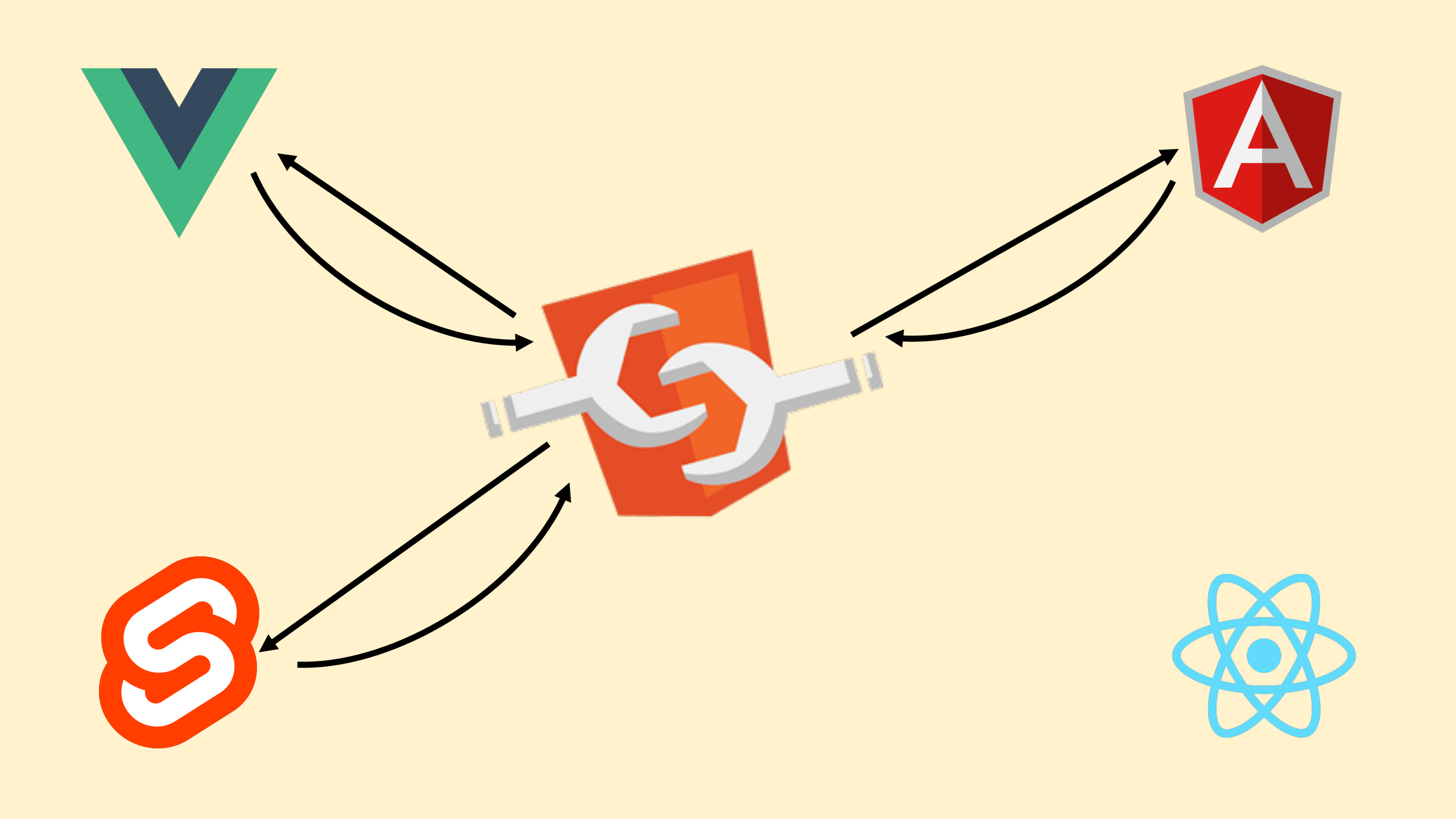
```
        :host([theme="danger"]) {
            --bg: red!important;
            --text: white!important;
        }
        :host([theme="warn"]) {
            --bg: gold!important;
            --text: black!important;
        }
        svg {
            grid-column: 2 / 3;
        }
        button {
            grid-column: 4;
            grid-row: 1;
            justify-self: end;
            margin-right: 8px;
        }
        ::slotted(*) {
            grid-column: 3 / 4;
            margin: 0;
            max-width: 66ch;
            line-height: 1.6;
        }
    `
}
```

```
static get properties() {
    return {
        icon: { type: String },
        theme: { type: String },
    }
}

constructor() {
    super()
    this.setAttribute('role', 'alert’)
    this.setAttribute('aria-live', 'polite’)
}

_close() {
    this.hidden = true;
}

render() {
    return html`
        <svg xmlns="http://www.w3.org/2000/svg" style="display: none;" >
            <symbol id="note" viewBox="0 0 24 24" fill="none" stroke="currentColor">
                <path strokeLinecap="round" strokeLinejoin="round" strokeWidth="2" d="..." />
            </symbol>
            <symbol id="warn" viewBox="0 0 24 24" fill="none" stroke="currentColor">
```

```
    }

    render() {
        return html`
            <svg xmlns="http://www.w3.org/2000/svg" style="display: none;" >
                <symbol id="note" viewBox="0 0 24 24" fill="none" stroke="currentColor">
                    <path strokeLinecap="round" strokeLinejoin="round" strokeWidth="2" d="..." />
                </symbol>
                <symbol id="warn" viewBox="0 0 24 24" fill="none" stroke="currentColor">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="..." />
                </symbol>
            </svg>

            <svg part="icon" height="24" width="24"><use href="#${this.icon}"></use></svg>
            <slot></slot>
            <button part="button" @click="${this._close}">Close</button>
        `
    }
}

window.customElements.define('custom-alert', CustomAlert)
```

A basic example





Custom Elements Everywhere

https://custom-elements-everywhere.com

Guest

Custom Elements Everywhere

Making sure frameworks and custom elements can be BFFs 🍻

LIBRARY	SCORE	BASIC TESTS	ADVANCED TESTS
Angular 8.2.14	100%	16/16	14/14

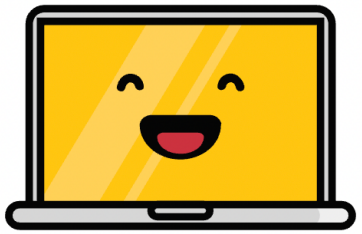
Handling data

Angular's default binding syntax will always set properties on an element. This works well for rich data, like objects and arrays, and also works well for primitive values so long as the Custom Element author has mapped any exposed attributes to corresponding properties.

Angular also provides binding syntax specifically for setting an attribute, if a developer would prefer to communicate with an element that way.

Handling events

Related Issues




Google


Microsoft


IBM


YouTube

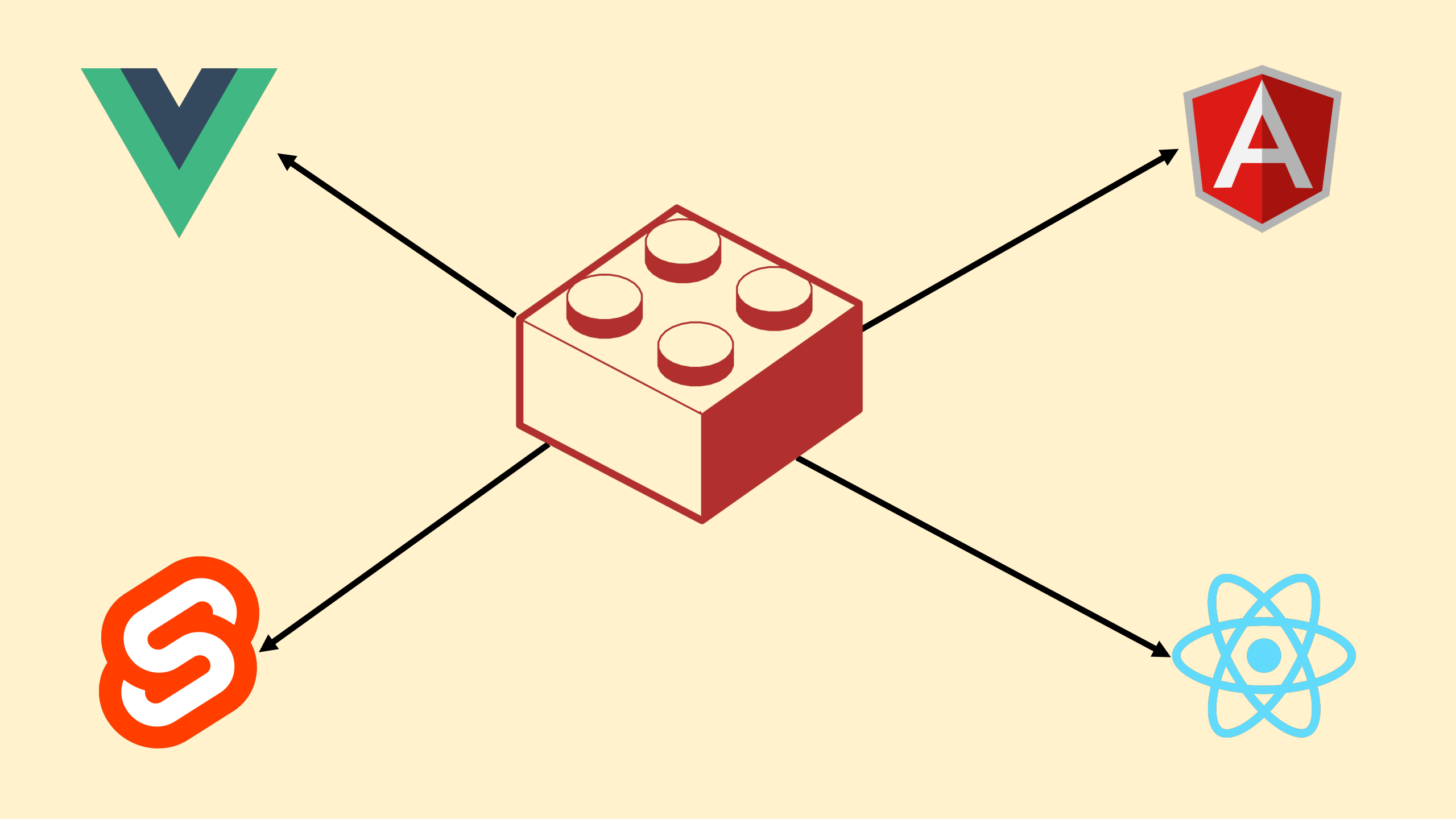

GitHub

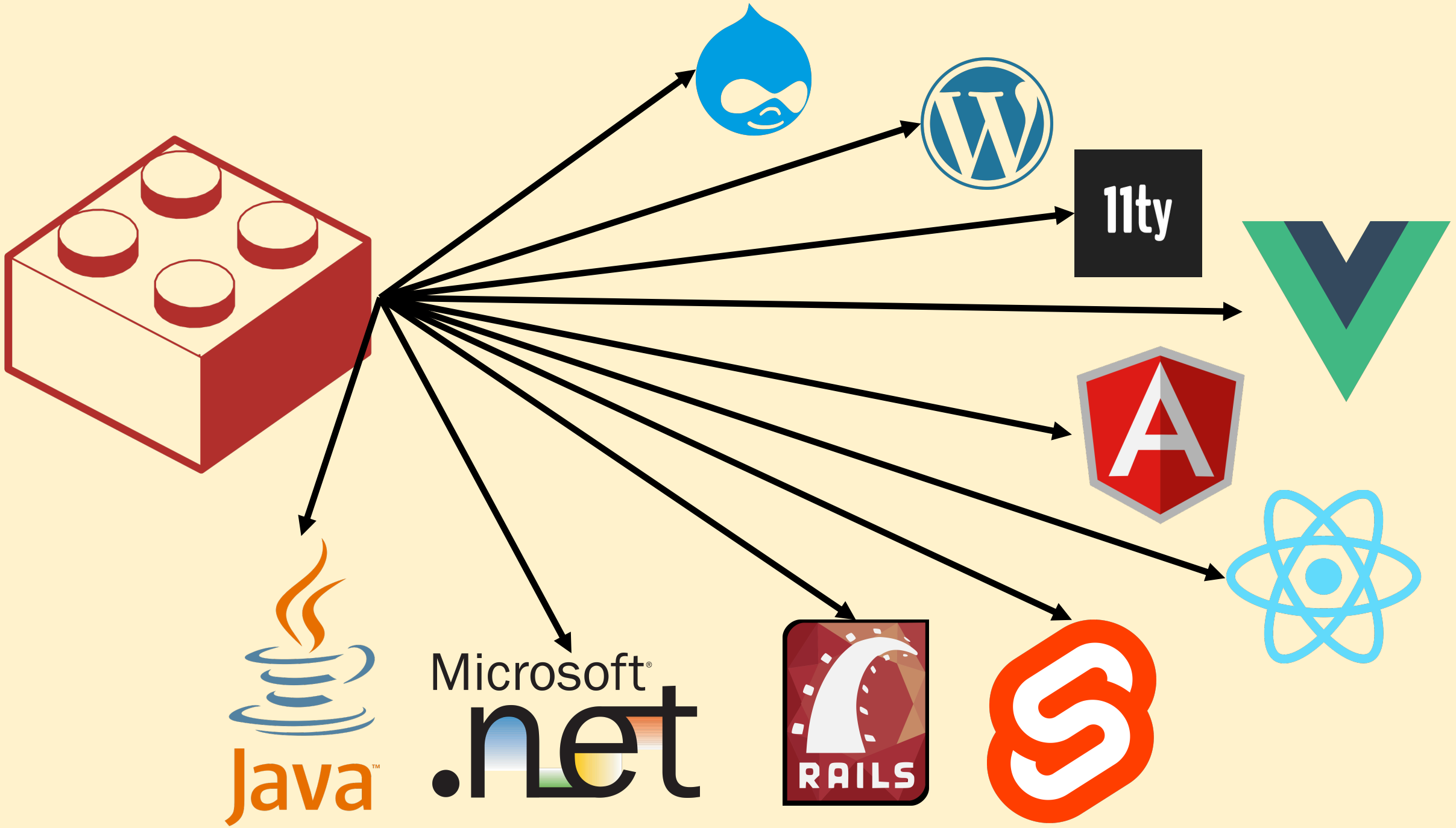

Red Hat


salesforce


ING


Adobe

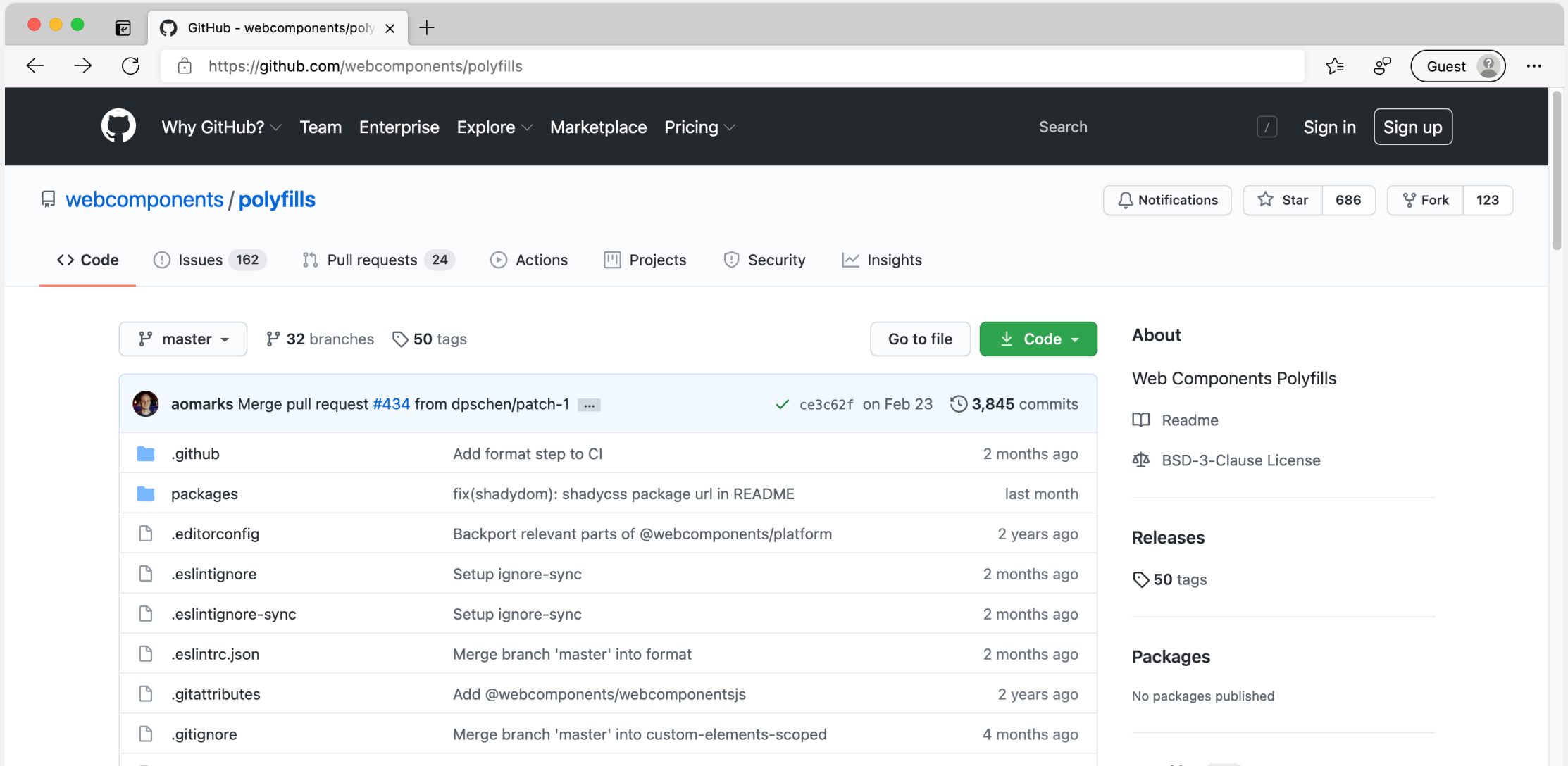



11ty
Java
Microsoft
.net
RAILS

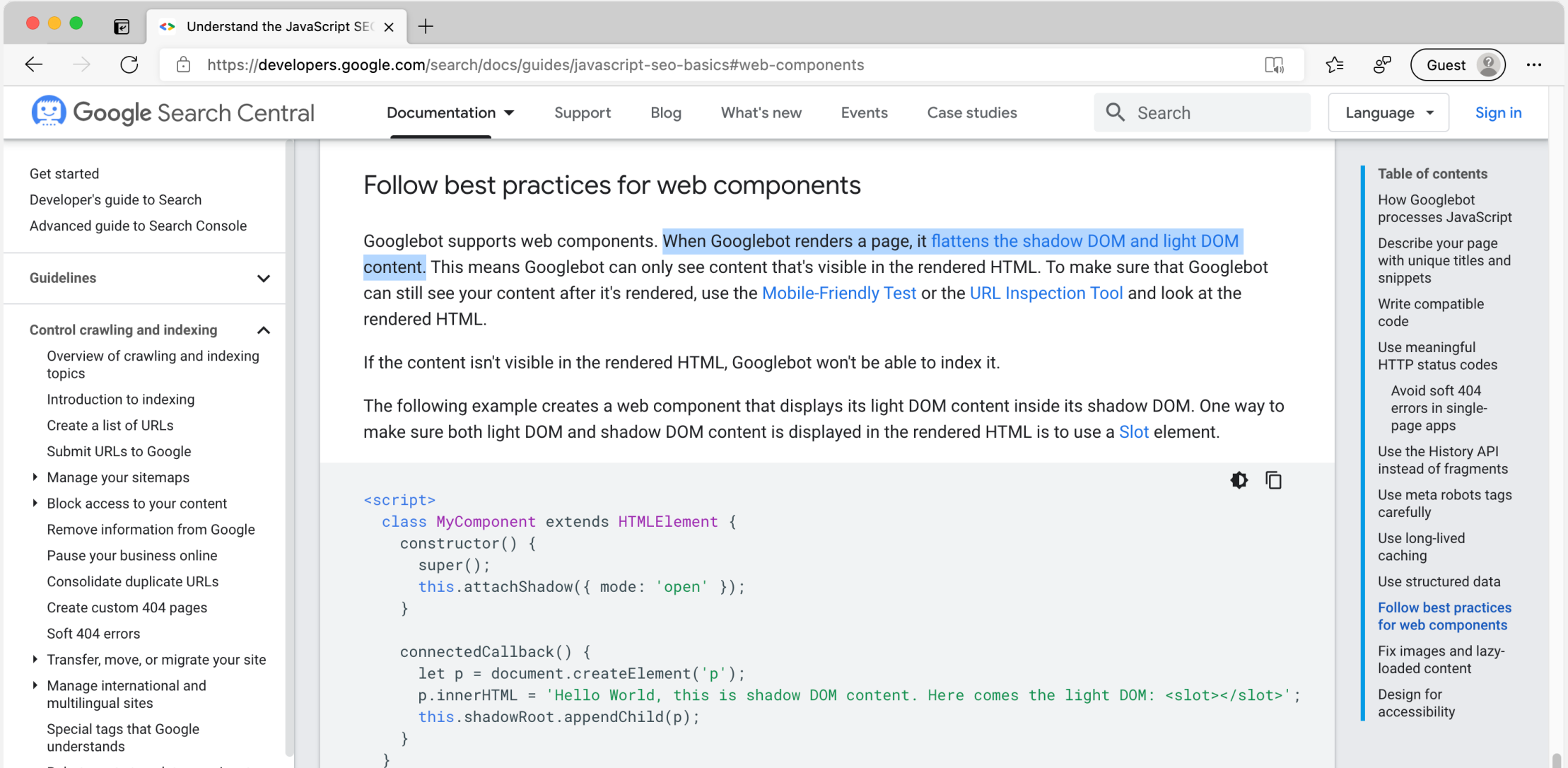

11ty
Java
Microsoft
.net
RAILS

FAQ

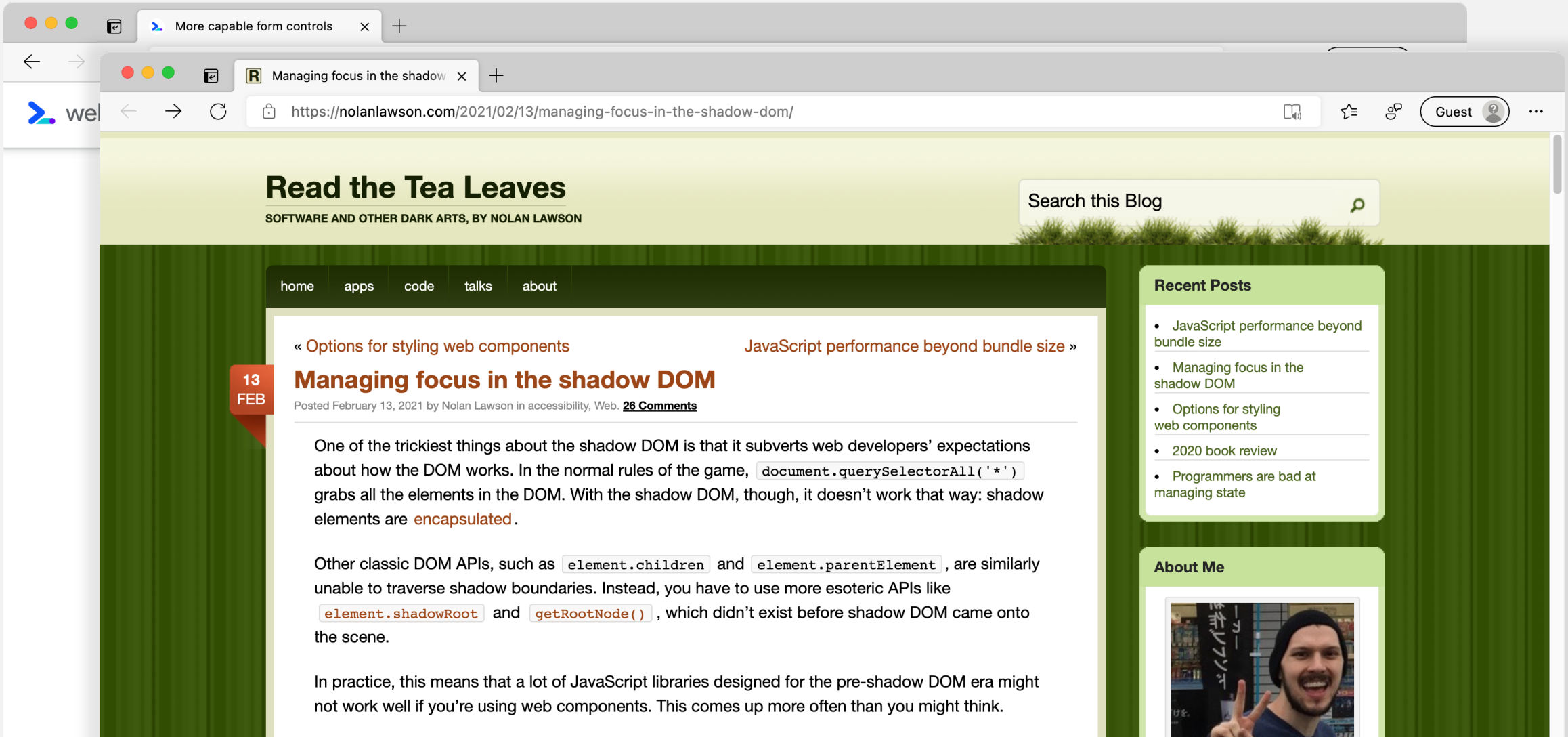
IE11?


GitHub - webcomponents/poly
https://github.com/webcomponents/polyfills
Guest
Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
webcomponents / polyfills
Notifications
Star 686
Fork 123
Code
Issues 162
Pull requests 24
Actions
Projects
Security
Insights
master
32 branches
50 tags
Go to file
Code
aomarks Merge pull request #434 from dpschen/patch-1
ce3c62f on Feb 23
3,845 commits
.github Add format step to CI 2 months ago
packages fix(shadydom): shadycss package url in README last month
.editorconfig Backport relevant parts of @webcomponents/platform 2 years ago
.eslintignore Setup ignore-sync 2 months ago
.eslintignore-sync Setup ignore-sync 2 months ago
.eslintrc.json Merge branch 'master' into format 2 months ago
.gitattributes Add @webcomponents/webcomponentsjs 2 years ago
.gitignore Merge branch 'master' into custom-elements-scoped 4 months ago
About
Web Components Polyfills
Readme
BSD-3-Clause License
Releases
50 tags
Packages
No packages published

SEO?


Understand the JavaScript SE
https://developers.google.com/search/docs/guides/javascript-seo-basics#web-components
Guest
Google Search Central
Documentation
Support
Blog
What's new
Events
Case studies
Search
Language
Sign in
Get started
Developer's guide to Search
Advanced guide to Search Console
Guidelines
Control crawling and indexing
Overview of crawling and indexing topics
Introduction to indexing
Create a list of URLs
Submit URLs to Google
Manage your sitemaps
Block access to your content
Remove information from Google
Pause your business online
Consolidate duplicate URLs
Create custom 404 pages
Soft 404 errors
Transfer, move, or migrate your site
Manage international and multilingual sites
Special tags that Google understands
Follow best practices for web components
Googlebot supports web components. When Googlebot renders a page, it flattens the shadow DOM and light DOM content. This means Googlebot can only see content that's visible in the rendered HTML. To make sure that Googlebot can still see your content after it's rendered, use the Mobile-Friendly Test or the URL Inspection Tool and look at the rendered HTML.
If the content isn't visible in the rendered HTML, Googlebot won't be able to index it.
The following example creates a web component that displays its light DOM content inside its shadow DOM. One way to make sure both light DOM and shadow DOM content is displayed in the rendered HTML is to use a Slot element.
<script>
class MyComponent extends HTMLElement {
constructor() {
super();
this.attachShadow({ mode: 'open' });
}
connectedCallback() {
let p = document.createElement('p');
p.innerHTML = 'Hello World, this is shadow DOM content. Here comes the light DOM: <slot></slot>';
this.shadowRoot.appendChild(p);
}
}
Table of contents
How Googlebot processes JavaScript
Describe your page with unique titles and snippets
Write compatible code
Use meaningful HTTP status codes
Avoid soft 404 errors in single-page apps
Use the History API instead of fragments
Use meta robots tags carefully
Use long-lived caching
Use structured data
Follow best practices for web components
Fix images and lazy-loaded content
Design for accessibility

A11y?


More capable form controls
Managing focus in the shadow
https://nolanlawson.com/2021/02/13/managing-focus-in-the-shadow-dom/
Guest
Read the Tea Leaves
SOFTWARE AND OTHER DARK ARTS, BY NOLAN LAWSON
Search this Blog
home
apps
code
talks
about
« Options for styling web components
JavaScript performance beyond bundle size »
13 FEB
Managing focus in the shadow DOM
Posted February 13, 2021 by Nolan Lawson in accessibility, Web. 26 Comments
One of the trickiest things about the shadow DOM is that it subverts web developers' expectations about how the DOM works. In the normal rules of the game, document.querySelectorAll('*') grabs all the elements in the DOM. With the shadow DOM, though, it doesn't work that way: shadow elements are encapsulated.
Other classic DOM APIs, such as element.children and element.parentElement, are similarly unable to traverse shadow boundaries. Instead, you have to use more esoteric APIs like element.shadowRoot and getRootNode(), which didn't exist before shadow DOM came onto the scene.
In practice, this means that a lot of JavaScript libraries designed for the pre-shadow DOM era might not work well if you're using web components. This comes up more often than you might think.
Recent Posts
JavaScript performance beyond bundle size
Managing focus in the shadow DOM
Options for styling web components
2020 book review
Programmers are bad at managing state
About Me

tl;dr

- Custom Elements around form controls have some gotchas.
  - Getting better with `ElementInternals` and `formdata`
- Focus management is tricky, like always.
  - `inert` may be a hero here.

What about my favorite things?

- State management?
- Server-side rendering?
- Props?
- Hot Module Replacement?

A few big ideas before we go...

Web Components build into the Web Platform

If we have good, accessible Custom Elements...

We have a pathway forward for more native elements

<tabs>, <accordion>, <model>, <popup>, etc.

Component Name Matrix

This table shows names of components from various UI works, self cross referenced with each other. Convergence is indicated by color and percentage.

Sort By: MATCH_COUNT

Match	Ant Des…	Atlaskit	Auth0 S…	Bootstrap	Carbon…	Chromi…	Evergr…	UI Fabric	FAST	Lightni…	Lion	Materia…	Primer …	Seman…	Stardus…	WAI-ARIA	Web Pl…
94%	Button	Button	Buttons	Buttons	Button		Button	Button	Button	Buttons	button	Buttons	Button	Button	Button	Button	Button
76%	Checkbox	Checkbox		Checkbox	Checkbox		Checkbox	Checkbox	Checkbox	Checkbox		Checkbox		Checkbox	Checkbox	Checkbox	Checkbox
76%	Tooltip	Tooltip		Tooltip	Tooltip		Tooltip	Tooltip	Tooltip	Tooltip	tooltip	Tooltip	Tooltip	Tooltip	Tooltip		
59%	Tabs	Tabs			Tabs		Tab			Tabs	tabs	Tabs		Tab		Tabs	
53%	Breadcr…	Breadcr…	Breadcr…	Breadcr…	Breadcr…			Breadcr…		Breadcr…				Breadcr…		Breadcr…	
47%	Form	Form	Forms	Forms	Form						form			Form	Form		
47%	Slider				Slider			Slider	Slider	Slider		Sliders			Slider	Slider	
47%	Badge	Badge	Badges	Badge			Badge		Badge	Badges		Badge					
47%	Card		Cards	Card					Card	Cards		Cards		Card	Card		
41%	Menu						Menu		Menu	Menus		Menus		Menu	Menu		
41%	Select	Select			Select		Select			Select	select						Select
41%	Alert		Alerts	Alerts			Alert			Alert					Alert	Alert	
41%	Modal		Modals	Modal	Modal			Modal		Modals				Modal			

An easier Web lowers barriers

Less complexity, less gatekeeping, a web for everyone.

Thanks

Dave Rupert @davatron5000