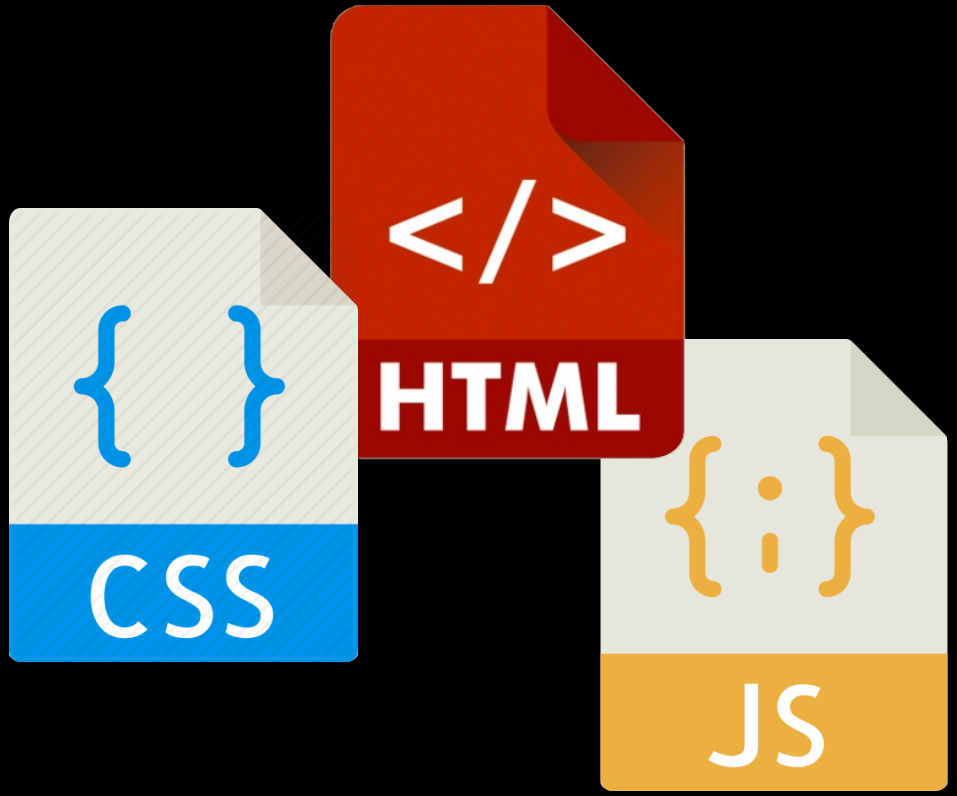
LitElement and Lit-HTML

To infinity and beyond!

Alaska AIRLINES

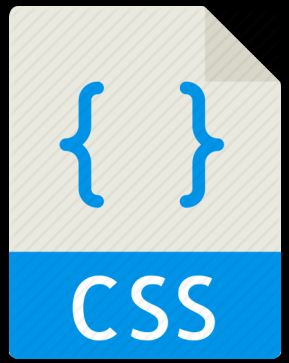

OPINIONS...
OPINIONS EVERYWHERE.
makeameme

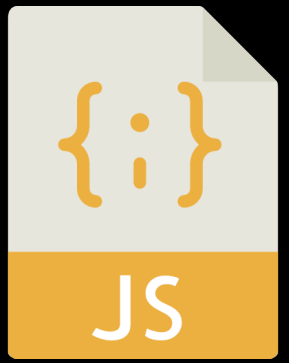

{ }
CSS
</>
HTML
{:}
JS

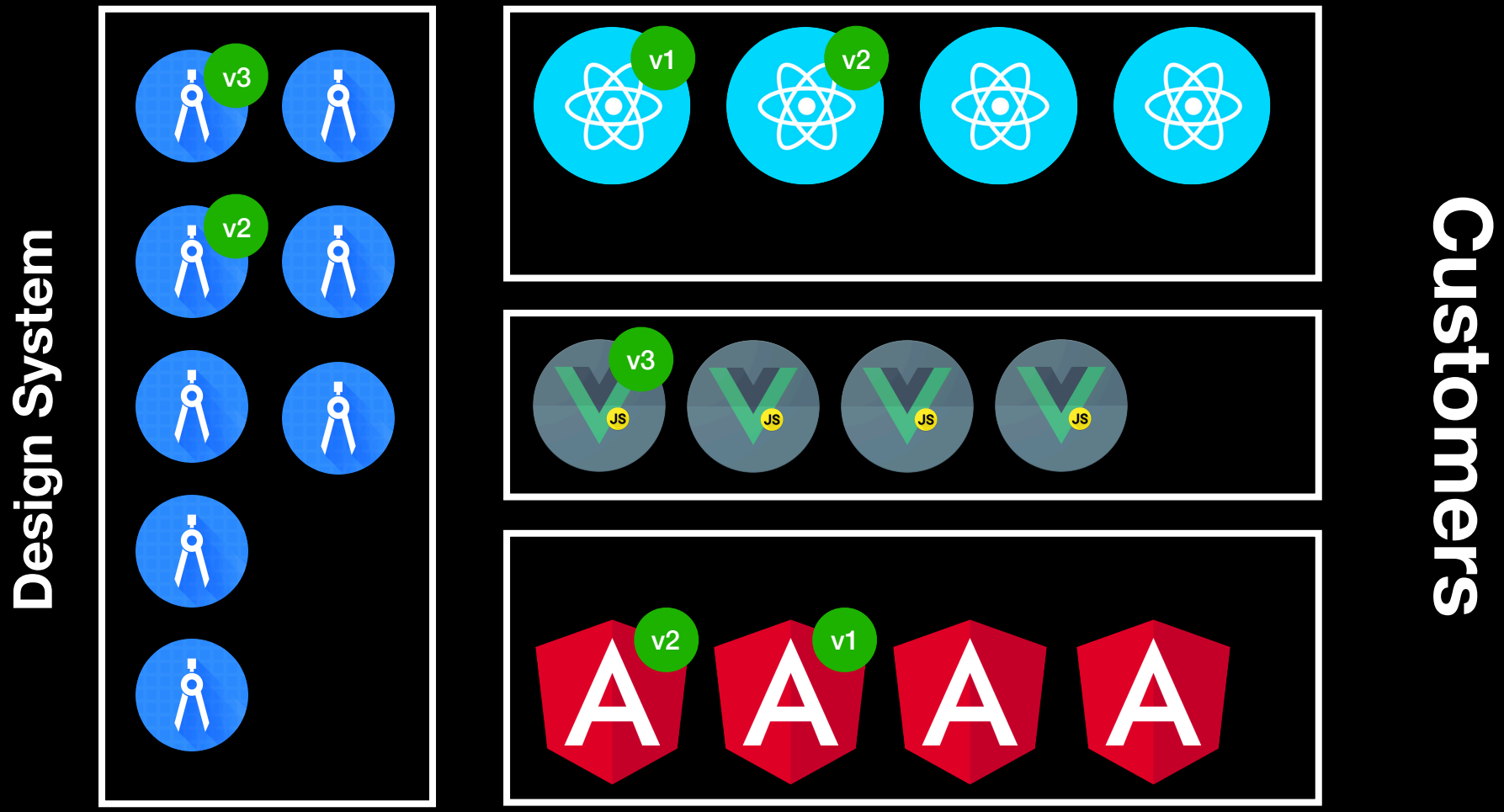
FASHION TRENDS

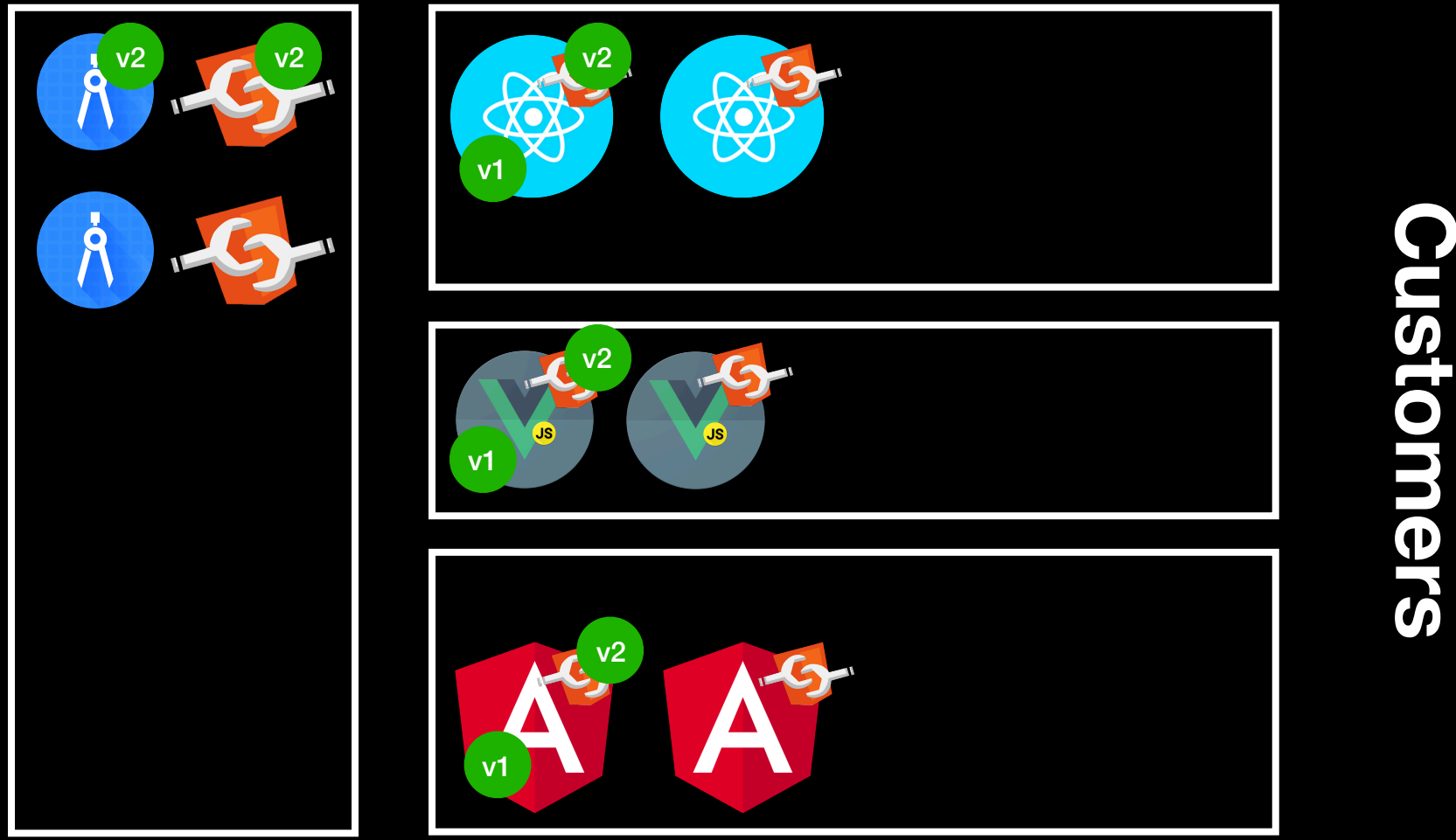
Some Are Just Stupid.


HTML


CSS


JS


Design System
v3
v2
v1
v2
v3
JS
JS
JS
JS
v2
v1
A
A
A
A
Customers


Design System
v2
v2
v2
v1
v2
v1
v2
v1
Customers

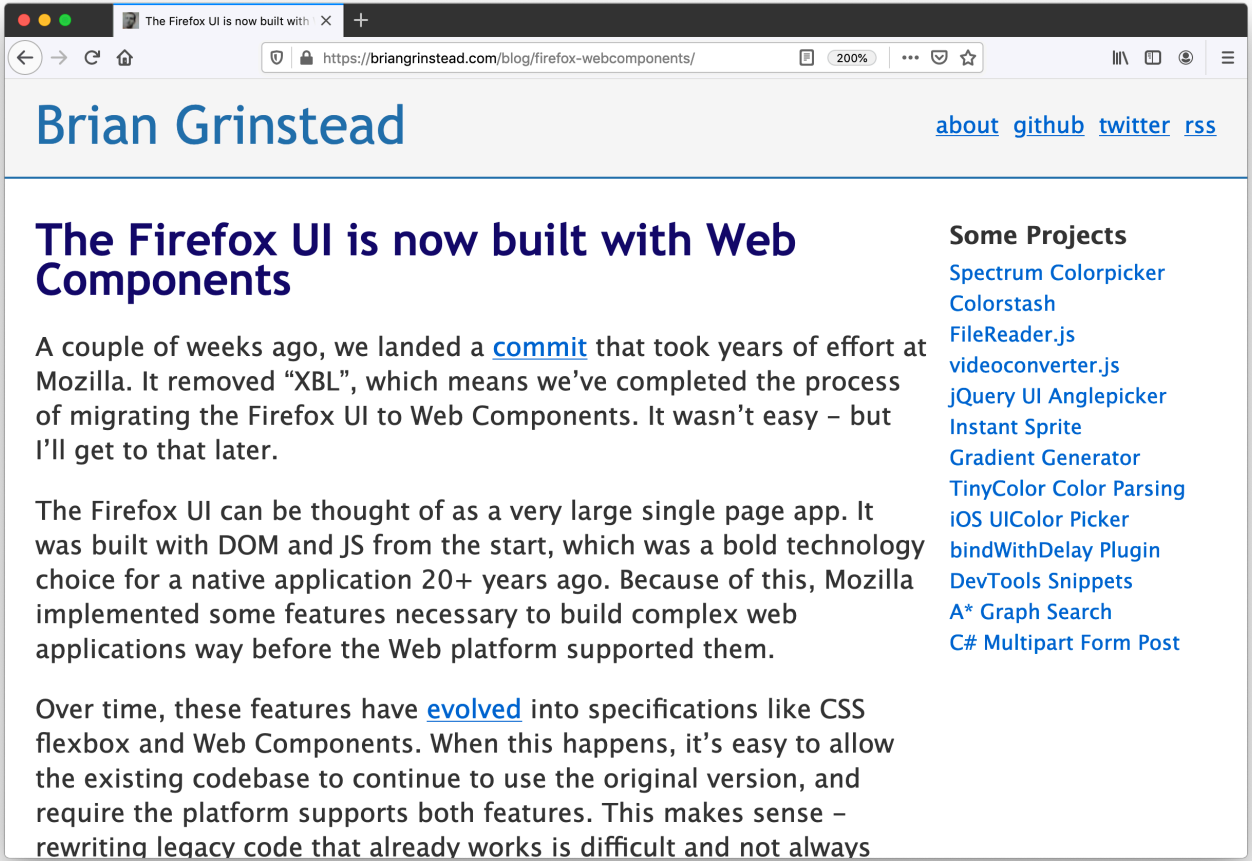
First there was Polymer …

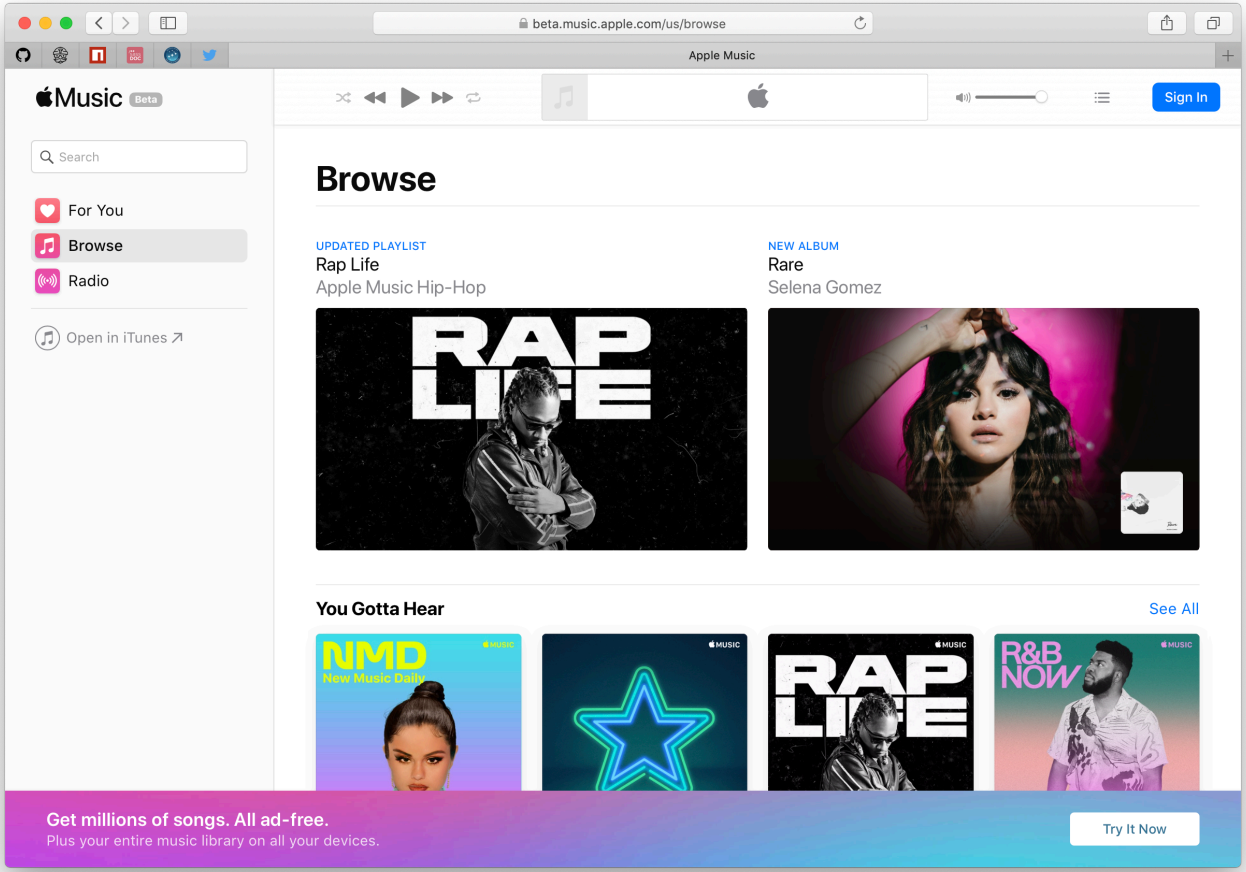
- 2015-05-29: Polymer 1.0 released
- 2015-09-24: Build components with ES6 classes
- 2016-03-10: Community Highlights - GE Predix UI
- 2016-05-19: Polymer at Google I/O 2016
- 2017-01-09: webcomponents.org launched
- 2017-05-15: Polymer 2.0 released
- 2018-05-09: Polymer 3.0 released
- 2018-12-13: lit-html 1.0 released
- 2019-01-11: litElement released

Today … major software companies have released tools based on Web Component technologies. Apple, Microsoft, and Google of course have made major investments into this tech!

Alaska AIRLINES

2019-02-05	Lightning-fast templates & Web Components: lit-html & LitElement Justin Fagnani
2019-01-11	LitElement release candidate Justin Fagnani
2018-12-13	lit-html 1.0 release candidate Gray Norton
2018-10-22	Latest releases from the Polymer Project An update on the latest releases from the Polymer Project. Polymer Team
2018-10-02	Web Components v0 deprecations With new Web Components specs shipping widely, older web components standards are being deprecated and removed from Chrome. Polymer Team
2018-05-25	Polymer Elements 3.0 FAQ FAQ about updating your application to use the Polymer Elements 3.0 Polymer Team
2018-05-09	Polymer @ Google I/O 2018 Polymer 3.0 released, next-generation products unveiled. Polymer Team
2018-05-02	Roadmap update, part 2: FAQ Frequently-asked-questions on our roadmap update. Gray Norton
2018-05-02	Roadmap update, part 1: 3.0 and beyond Updates on the Polymer 3.0 release and what comes


The Firefox UI is now built with
https://briangrinstead.com/blog/firefox-webcomponents/
200%
Brian Grinstead
about github twitter rss
The Firefox UI is now built with Web Components
A couple of weeks ago, we landed a commit that took years of effort at Mozilla. It removed "XBL", which means we've completed the process of migrating the Firefox UI to Web Components. It wasn't easy – but I'll get to that later.
The Firefox UI can be thought of as a very large single page app. It was built with DOM and JS from the start, which was a bold technology choice for a native application 20+ years ago. Because of this, Mozilla implemented some features necessary to build complex web applications way before the Web platform supported them.
Over time, these features have evolved into specifications like CSS flexbox and Web Components. When this happens, it's easy to allow the existing codebase to continue to use the original version, and require the platform supports both features. This makes sense – rewriting legacy code that already works is difficult and not always
Some Projects
Spectrum Colorpicker
Colorstash
FileReader.js
videoconverter.js
jQuery UI Anglepicker
Instant Sprite
Gradient Generator
TinyColor Color Parsing
iOS UIColor Picker
bindWithDelay Plugin
DevTools Snippets
A* Graph Search
C# Multipart Form Post


beta.music.apple.com/us/browse
Apple Music
Music Beta
Search
For You
Browse
Radio
Open in iTunes
Sign In
Browse
UPDATED PLAYLIST
Rap Life
Apple Music Hip-Hop
NEW ALBUM
Rare
Selena Gomez
You Gotta Hear
See All
NMD
New Music Daily
RAP LIFE
R&B NOW
Get millions of songs. All ad-free.
Plus your entire music library on all your devices.
Try It Now

What about IE?

IE11 is supported with a polyfill supported by web components.org … so, meh

Alaska AIRLINES

The future is now!

1. Chrome 
2. Safari 
3. Firefox 
4. Microsoft Edge  Official release 1.15.20

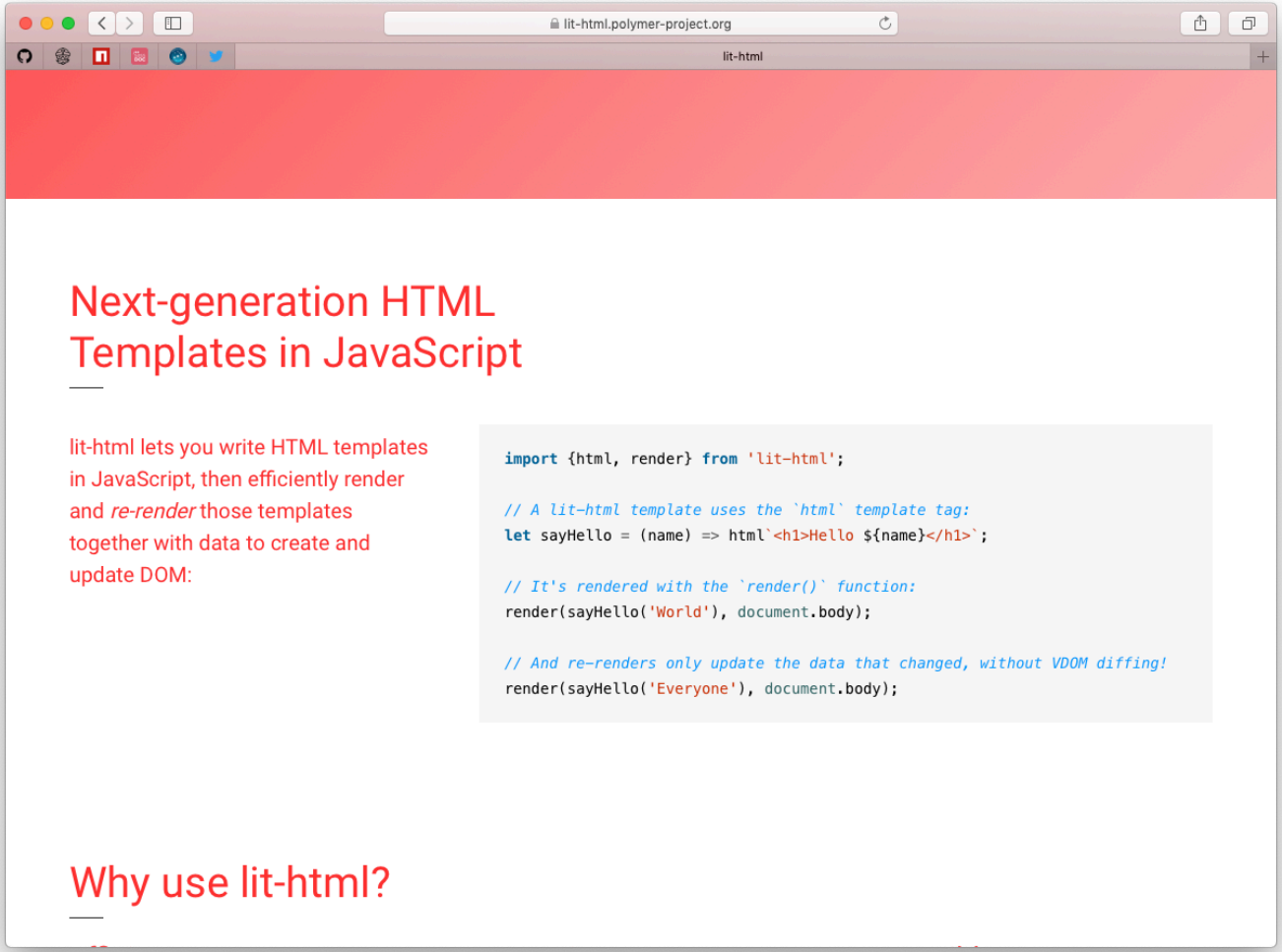
Alaska AIRLINES

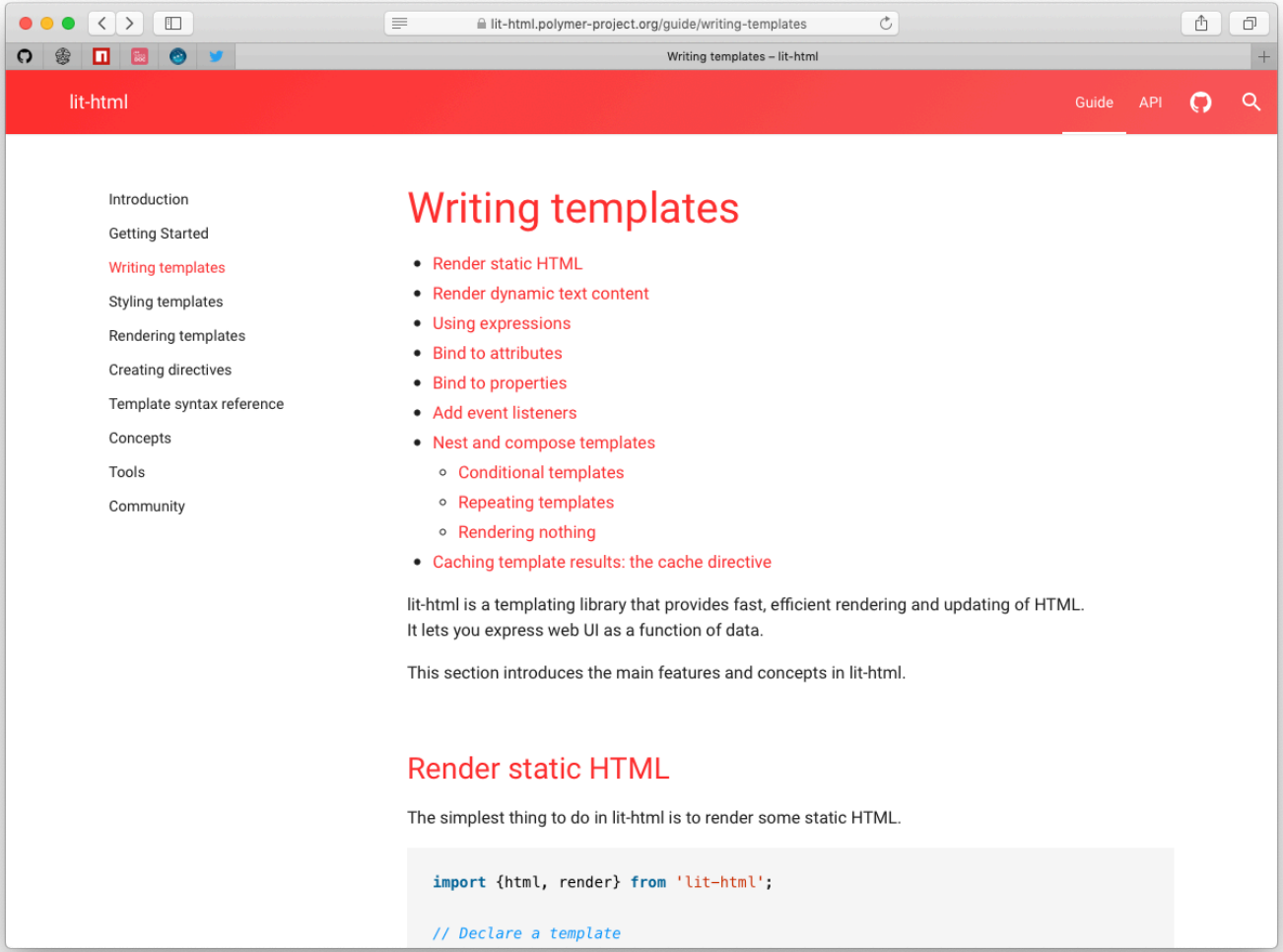
let's build our 1st component

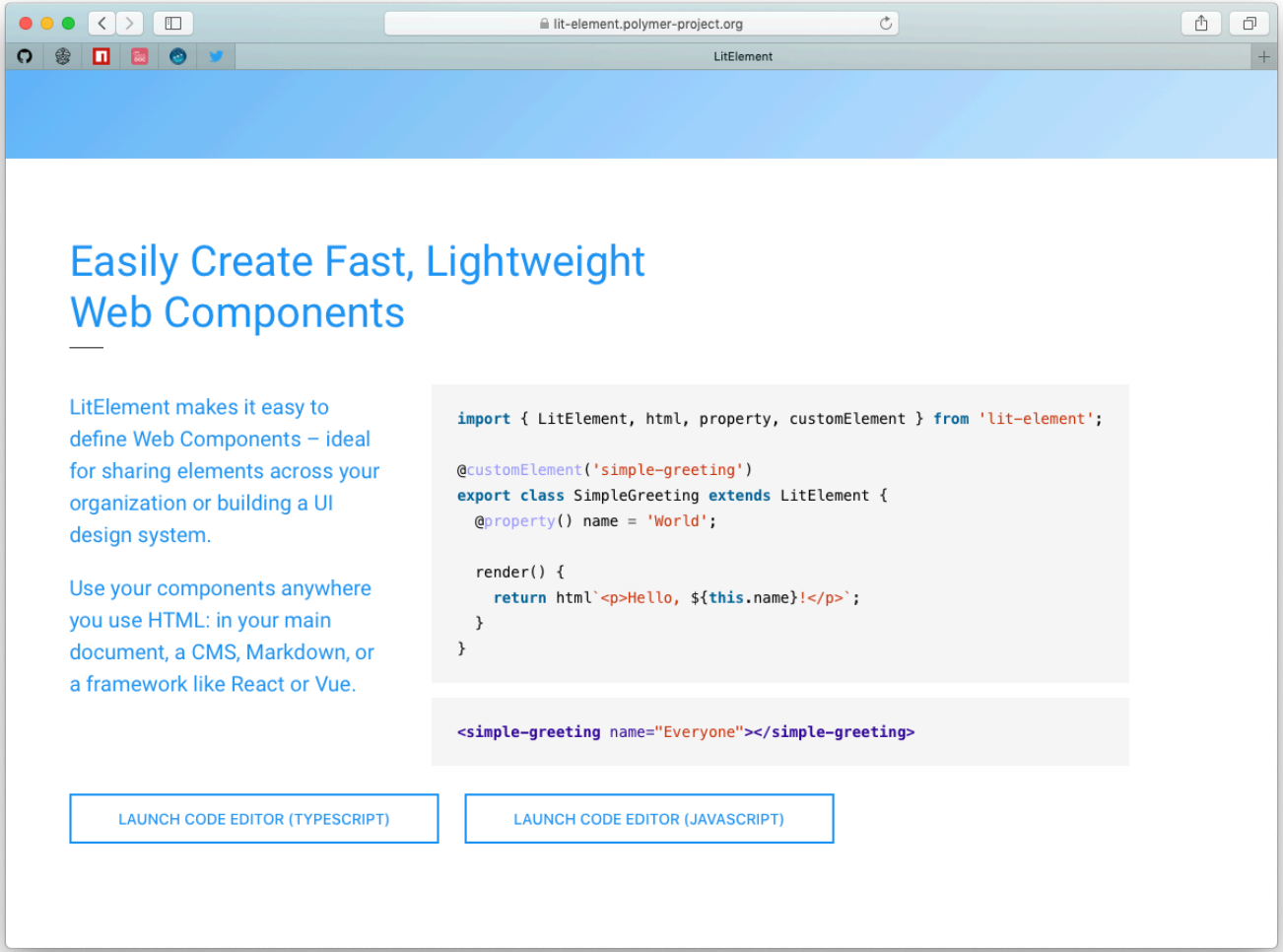
To build a web component you need a few things …

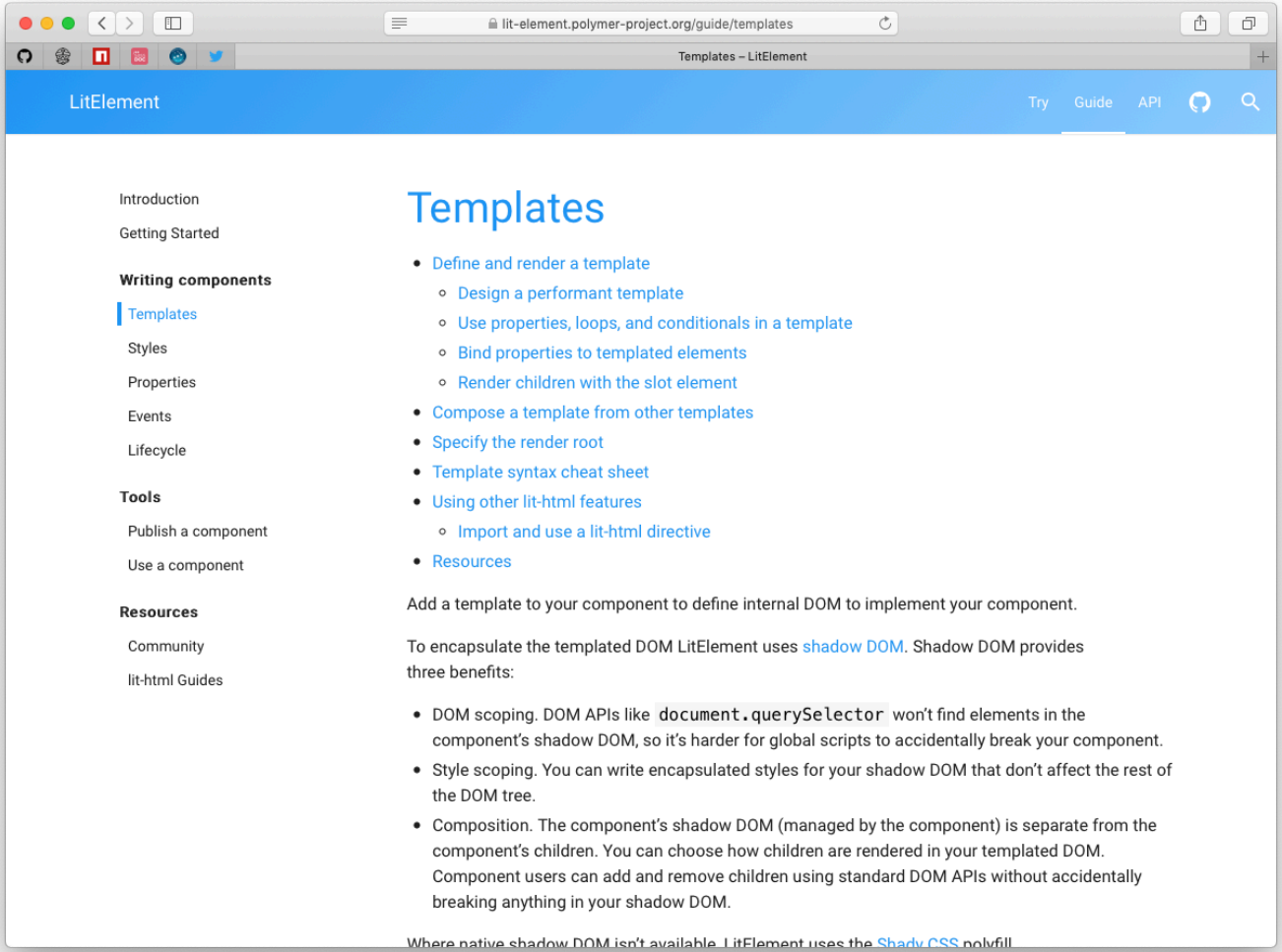
1. A computer
2. Web browser
3. Code editor

Alaska AIRLINES


lit-html.polymer-project.org
lit-html
Next-generation HTML Templates in JavaScript
lit-html lets you write HTML templates in JavaScript, then efficiently render and re-render those templates together with data to create and update DOM:
import {html, render} from 'lit-html';
// A lit-html template uses the `html` template tag:
let sayHello = (name) => html`<h1>Hello ${name}</h1>`;
// It's rendered with the `render()` function:
render(sayHello('World'), document.body);
// And re-renders only update the data that changed, without VDOM diffing!
render(sayHello('Everyone'), document.body);
Why use lit-html?


lit-html.polymer-project.org/guide/writing-templates
Writing templates – lit-html
lit-html
Guide
API
Introduction
Getting Started
Writing templates
Styling templates
Rendering templates
Creating directives
Template syntax reference
Concepts
Tools
Community
Writing templates
Render static HTML
Render dynamic text content
Using expressions
Bind to attributes
Bind to properties
Add event listeners
Nest and compose templates
Conditional templates
Repeating templates
Rendering nothing
Caching template results: the cache directive
lit-html is a templating library that provides fast, efficient rendering and updating of HTML.
It lets you express web UI as a function of data.
This section introduces the main features and concepts in lit-html.
Render static HTML
The simplest thing to do in lit-html is to render some static HTML.
import {html, render} from 'lit-html';
// Declare a template


lit-element.polymer-project.org
LitElement
Easily Create Fast, Lightweight Web Components
LitElement makes it easy to define Web Components – ideal for sharing elements across your organization or building a UI design system.
Use your components anywhere you use HTML: in your main document, a CMS, Markdown, or a framework like React or Vue.
import { LitElement, html, property, customElement } from 'lit-element';
@customElement('simple-greeting')
export class SimpleGreeting extends LitElement {
@property() name = 'World';
render() {
return html`<p>Hello, ${this.name}!</p>`;
}
}
<simple-greeting name="Everyone"></simple-greeting>
LAUNCH CODE EDITOR (TYPESCRIPT)
LAUNCH CODE EDITOR (JAVASCRIPT)



Alaska AIRLINES

let's build our 1st component

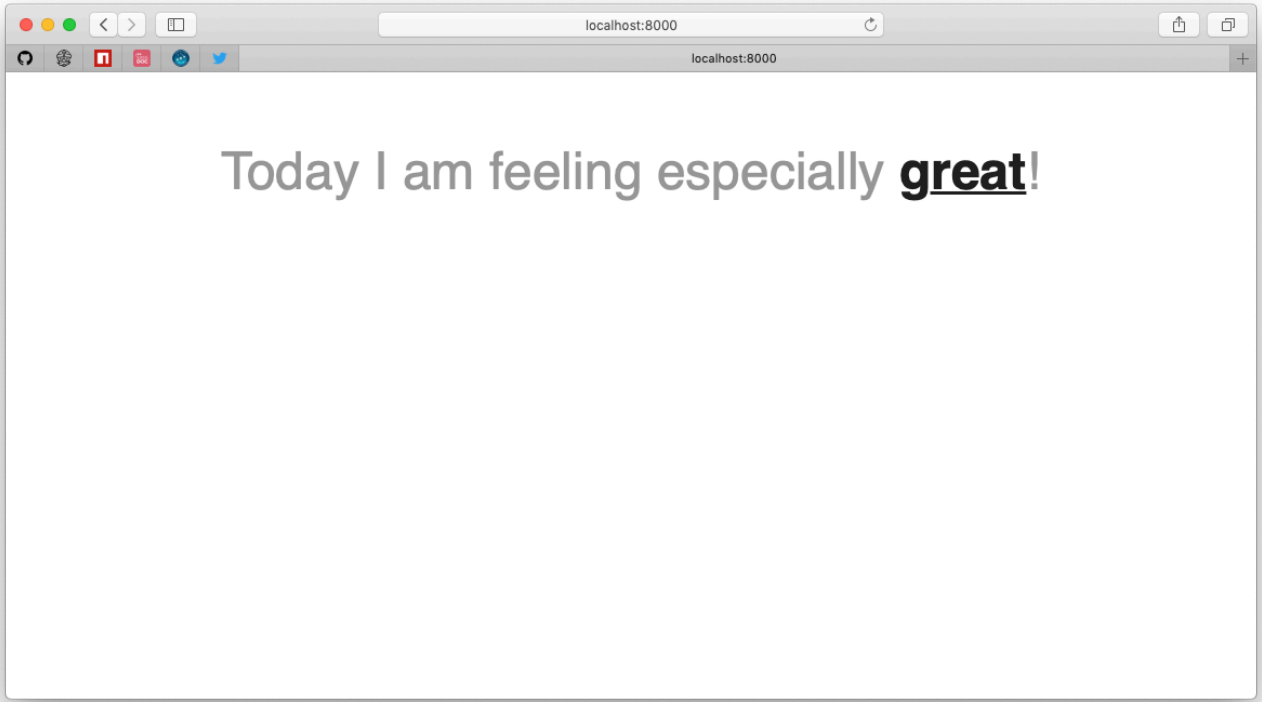
```
import {LitElement, html, css} from 'https://unpkg.com/lit-element/lit-element.js?module';

class MyMood extends LitElement {

  static get properties() {
    return {
      mood: { type: String }
    }
  }

  static get styles() {
    return css`
    :host {
      display: block;
      font-family: sans-serif;
      font-size: 3rem;
      color: #999;
      text-align: center;
      margin: 4rem 0;
    }
    span {
        text-decoration: underline;
        font-weight: bold;
        color: #222;
      }
    `;
  }

  render() {
    return html`
      Today I am feeling especially <span>${this.mood}</span>!
    `;
  }

}

customElements.define('my-mood', MyMood);

```

Alaska AIRLINES

let's use our 1st component

```
<html>
<head>
  <!-- Polyfills only needed for Firefox and Edge. -->
  <script
src="https://unpkg.com/@webcomponents/webcomponentsjs@latest/webcomponent
s-loader.js"></script>
</head>
<body>
  <script type="module" src="./my-mood.js"></script>

  <my-mood mood="great"></my-mood>
</body>
</html>

```

Alaska AIRLINES

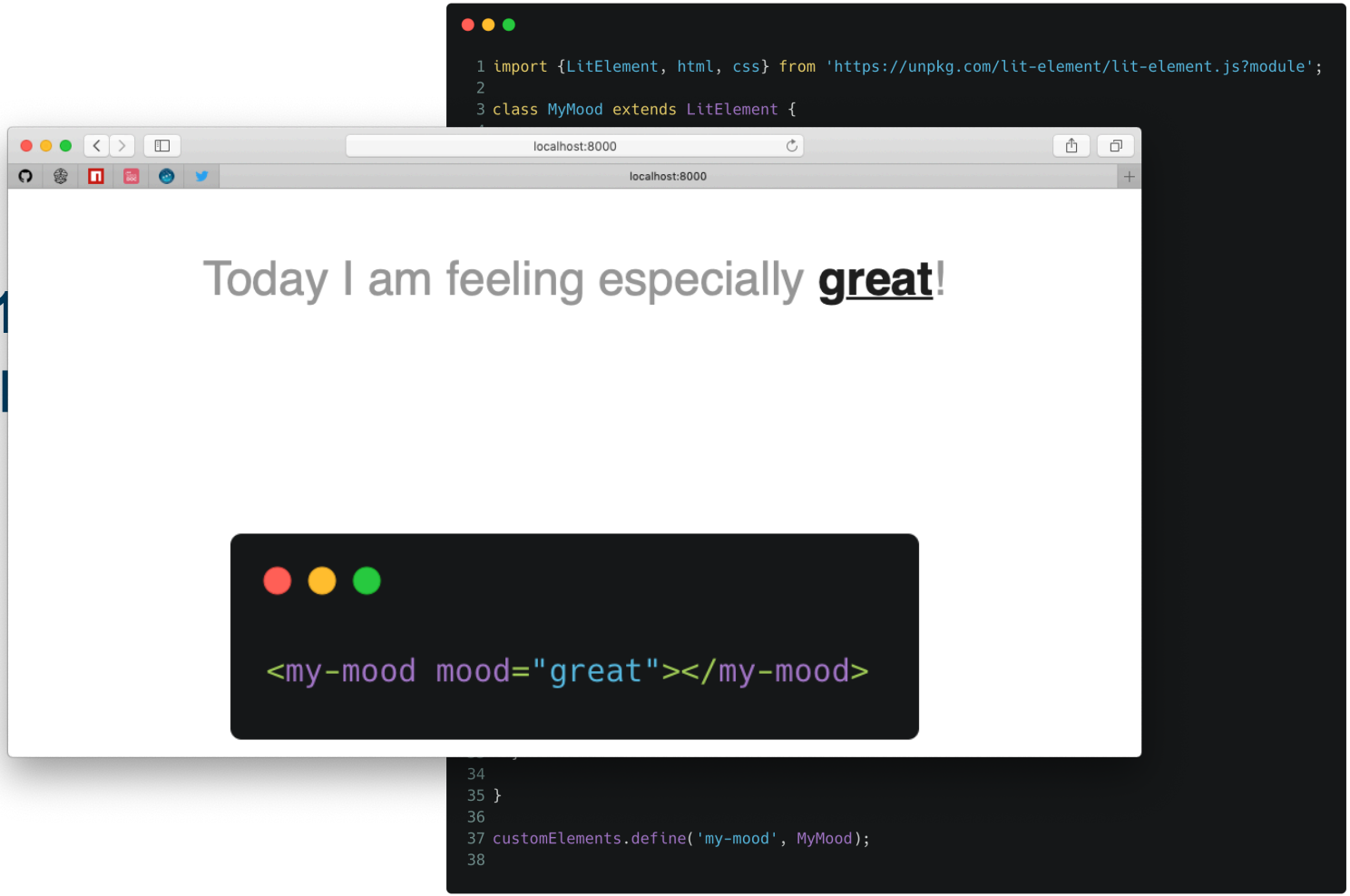

localhost:8000
localhost:8000
Today I am feeling especially great!

let's review our 1st component

```
import {LitElement, html, css} from 'https://unpkg.com/lit-element/lit-element.js?module';

class MyMood extends LitElement {

  static get properties() {
    return {
      mood: { type: String }
    }
  }

  static get styles() {
    return css`
    :host {
      display: block;
      font-family: sans-serif;
      font-size: 3rem;
      color: #999;
      text-align: center;
      margin: 4rem 0;
    }
    span {
        text-decoration: underline;
        font-weight: bold;
        color: #222;
      }
    `;
  }

  render() {
    return html`
      Today I am feeling especially <span>${this.mood}</span>!
    `;
  }

}

customElements.define('my-mood', MyMood);

```

Alaska AIRLINES

let's
our 1
comp



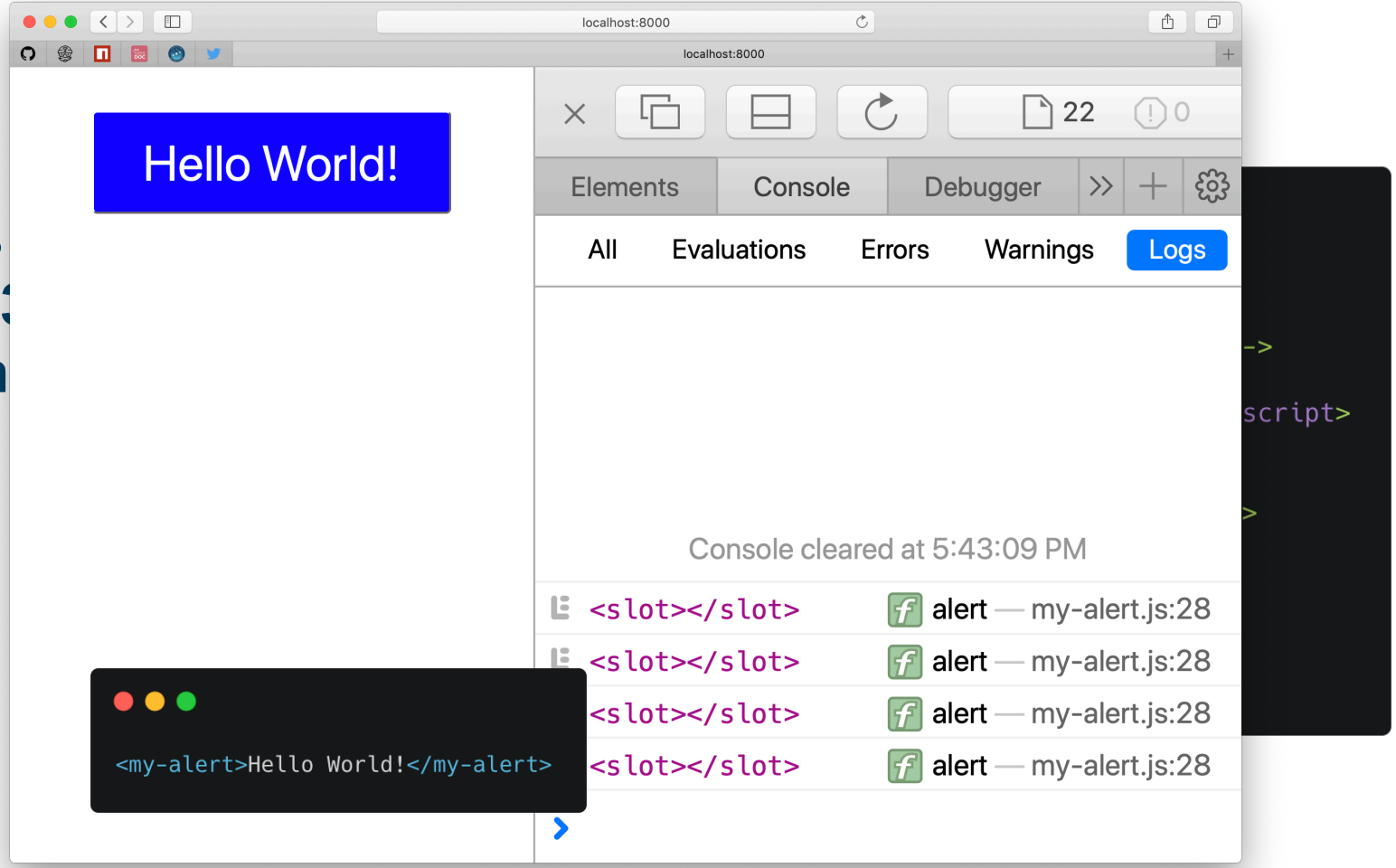
Alaska AIRLINES

let's build our 3rd component

```
import {LitElement, html, css} from 'https://unpkg.com/lit-element/lit-element.js?module';

class MyAlert extends LitElement {

  static get styles() {
    return css`
      :host {
        display: block;
        font-family: sans-serif;
        text-align: center;
        margin: 40px 0;
      }
      button {
        background-color: blue;
        padding: 0 1em;
        line-height: 2;
        border-radius: 5px;
        font-size: 3rem;
        color: #fff;
      }
      button:hover {
        cursor: hand;
      }
    `;
  }

  alert = (e) => {
    console.log(e.target);
  }

  render() {
    return html`
      <button @click="${this.alert}">
        <slot></slot>
      </button>
    `;
  }

}

customElements.define('my-alert', MyAlert);

```

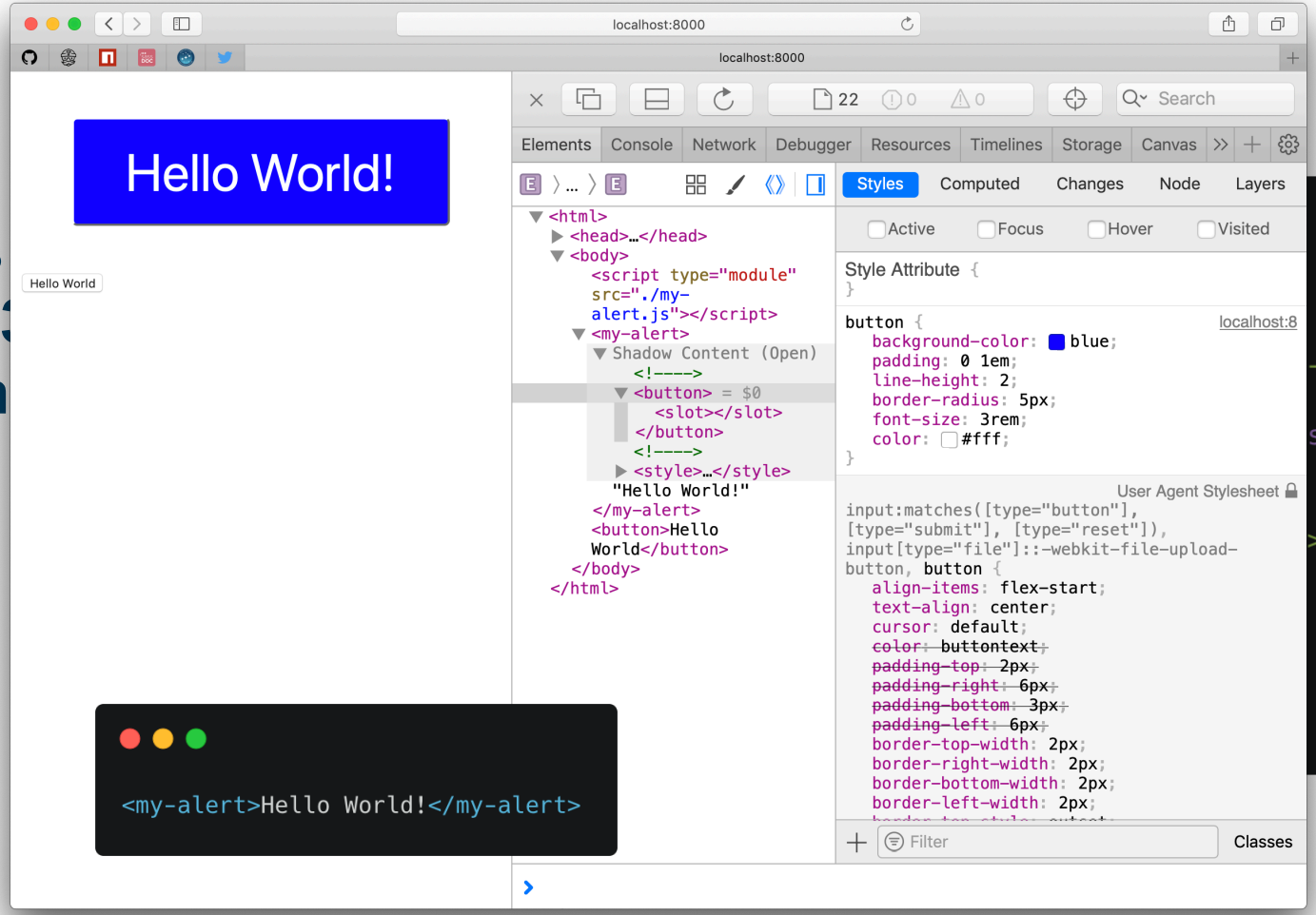
Alaska AIRLINES

let's
our
com

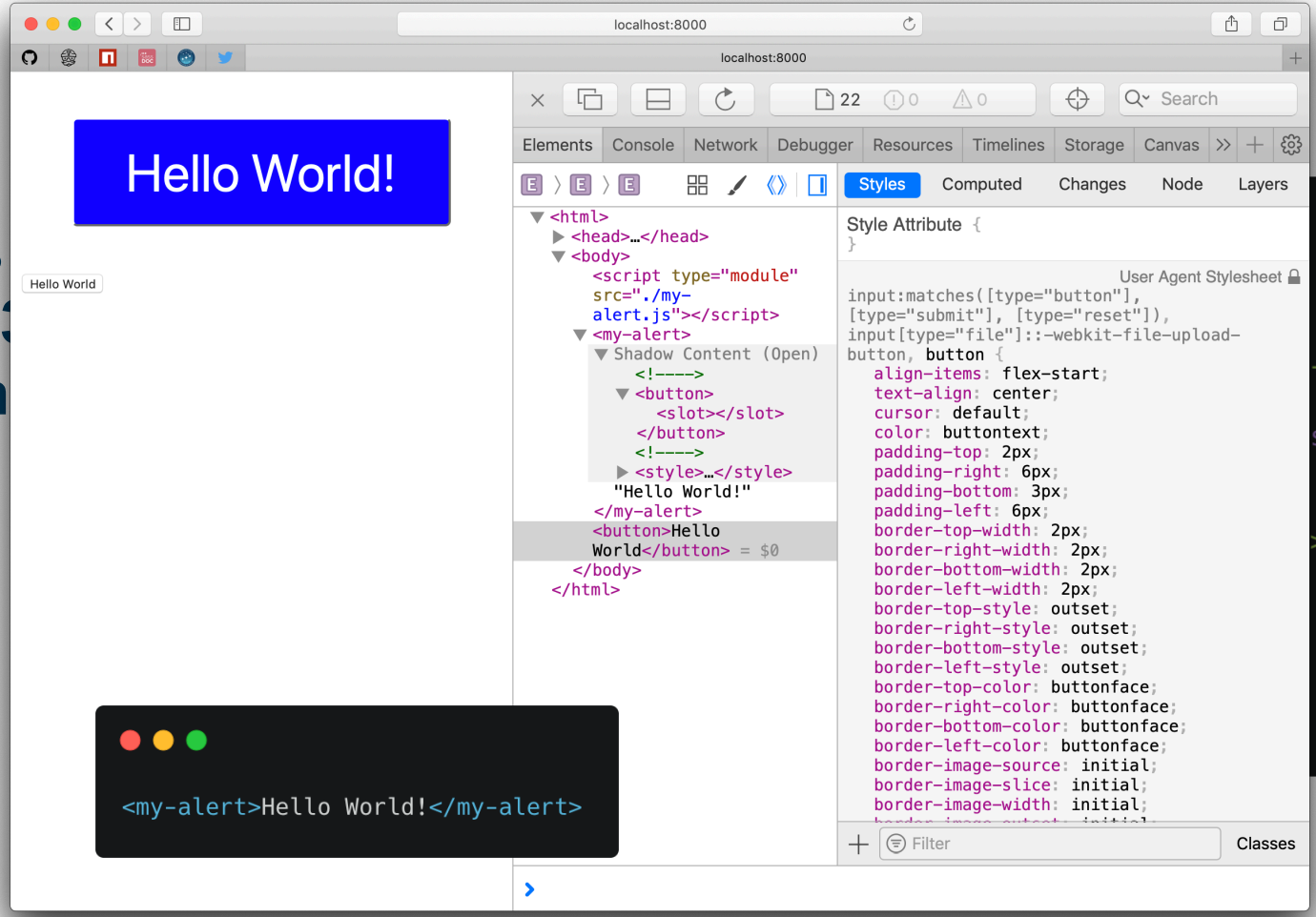


Alaska AIRLINES

let's
our
com


localhost:8000
localhost:8000
Hello World!
Hello World
Elements
Console
Network
Debugger
Resources
Timelines
Storage
Canvas
Search
Styles
Computed
Changes
Node
Layers
Active
Focus
Hover
Visited
Style Attribute {
}
button {
localhost:8
background-color: blue;
padding: 0 1em;
line-height: 2;
border-radius: 5px;
font-size: 3rem;
color: #fff;
}
User Agent Stylesheet
input:matches([type="button"], [type="submit"], [type="reset"]), input[type="file"]::-webkit-file-upload-button, button {
align-items: flex-start;
text-align: center;
cursor: default;
color: buttontext;
padding-top: 2px;
padding-right: 6px;
padding-bottom: 3px;
padding-left: 6px;
border-top-width: 2px;
border-right-width: 2px;
border-bottom-width: 2px;
border-left-width: 2px;
<my-alert>Hello World!</my-alert>
Filter
Classes

let's
our
com



Alaska AIRLINES

In closing …

1. HTML Web Components are ready for prime-time
2. WCs are being used in enterprise web apps by major companies
3. No frameworks or special tooling required
4. 2019 was a coming of age … 2020 is going to BLOW UP!

Thanks!

Alaska AIRLINES