



# Web Components From the Eyes of a Newcomer

**Brian Perry**

Front End Architect

**bounteous**





# Please abide by our Code of Conduct

All attendees, speakers, sponsors, and volunteers at our conference are required to agree with the our code of conduct.

We do not tolerate harassment of conference participants in any form.

# BRIAN PERRY

- » Front End Architect at Bounteous
- » Rocking the Chicago 'burbs
- » Lover of all things components... and Nintendo

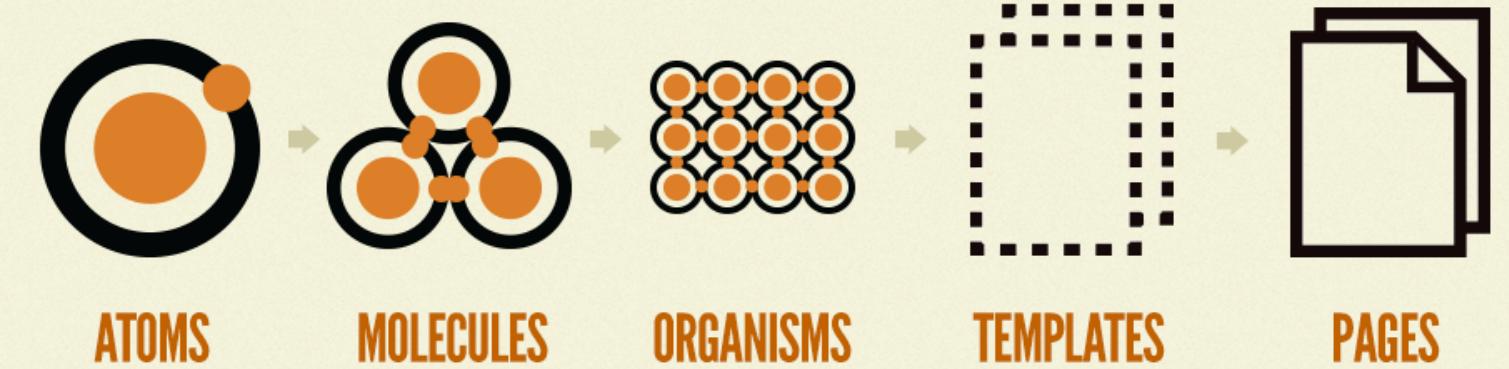
@bricomedy  
[brianperry.dev](http://brianperry.dev)



# bounteous



# COMPONENTS HAVE TAKEN OVER THE WEB



**WE'RE HERE TO TALK  
ABOUT  
WEB  
COMPONENTS**

**LIKE REACT AND ANGULAR, RIGHT?**

# WEB COMPONENTS

A set of web platform APIs, not tied to a specific framework

- » Custom elements
- » Shadow DOM
- » HTML Templates

I ❤️ COMPONENTS.

SO I SHOULD ❤️ WEB COMPONENTS.

SO WHY AREN'T THEY PART OF MY  
WORKFLOW? 

A black and white photograph of a man from the chest up. He has short dark hair and is looking slightly upwards and to his right with a neutral expression. He is wearing a dark-colored, long-sleeved button-down shirt with a fine, light-colored checkered pattern. His left hand is resting on his left shoulder, with his fingers partially hidden by the fabric. In the background, there is a small, potted plant with many green leaves, positioned behind his right shoulder.

**WARNING: NOT AN EXPERT**

# I LOOK TO:



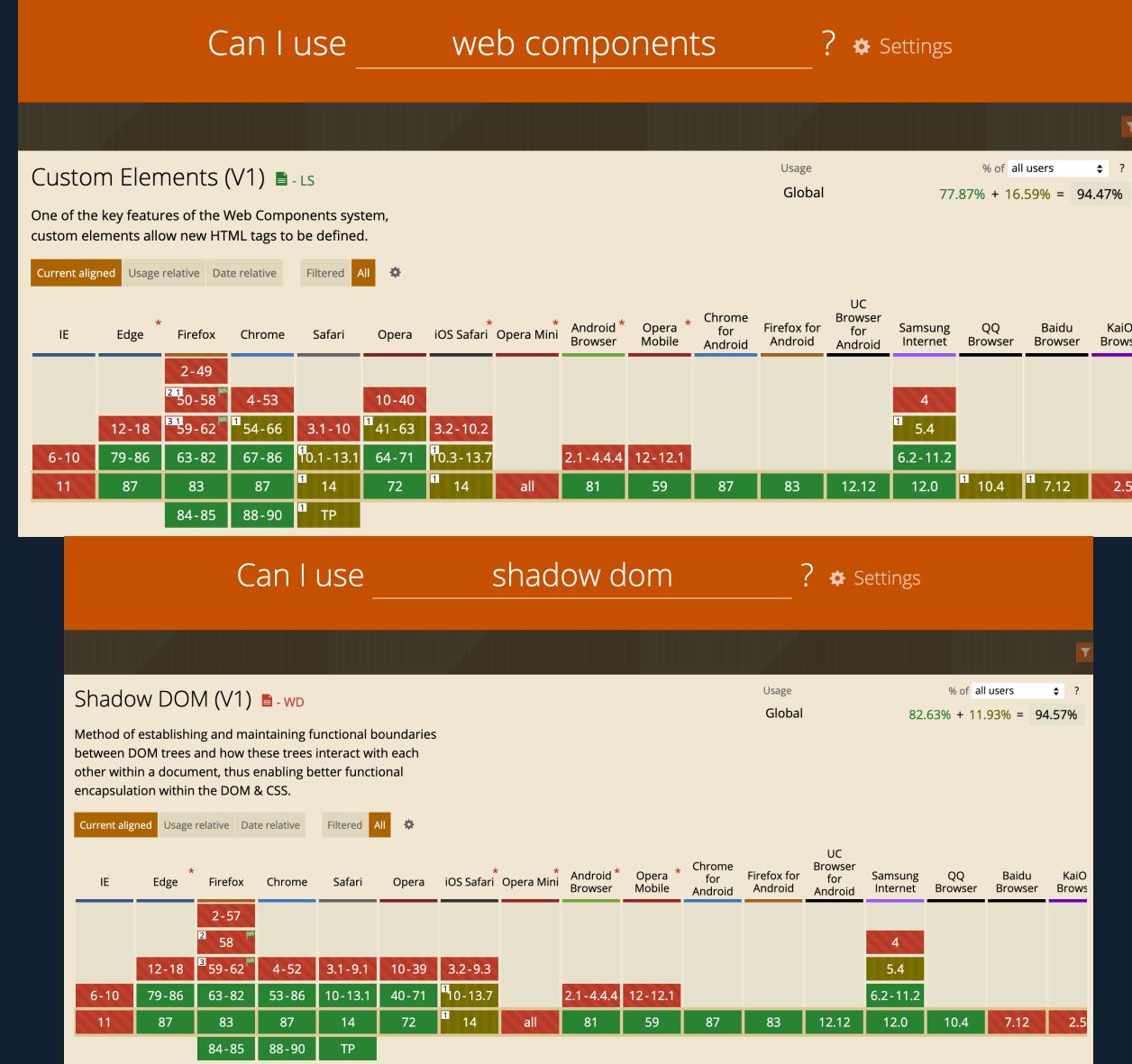
**@BTOPRO**

**@SALEM\_COBALT**

**@CASTASTROPHEE**

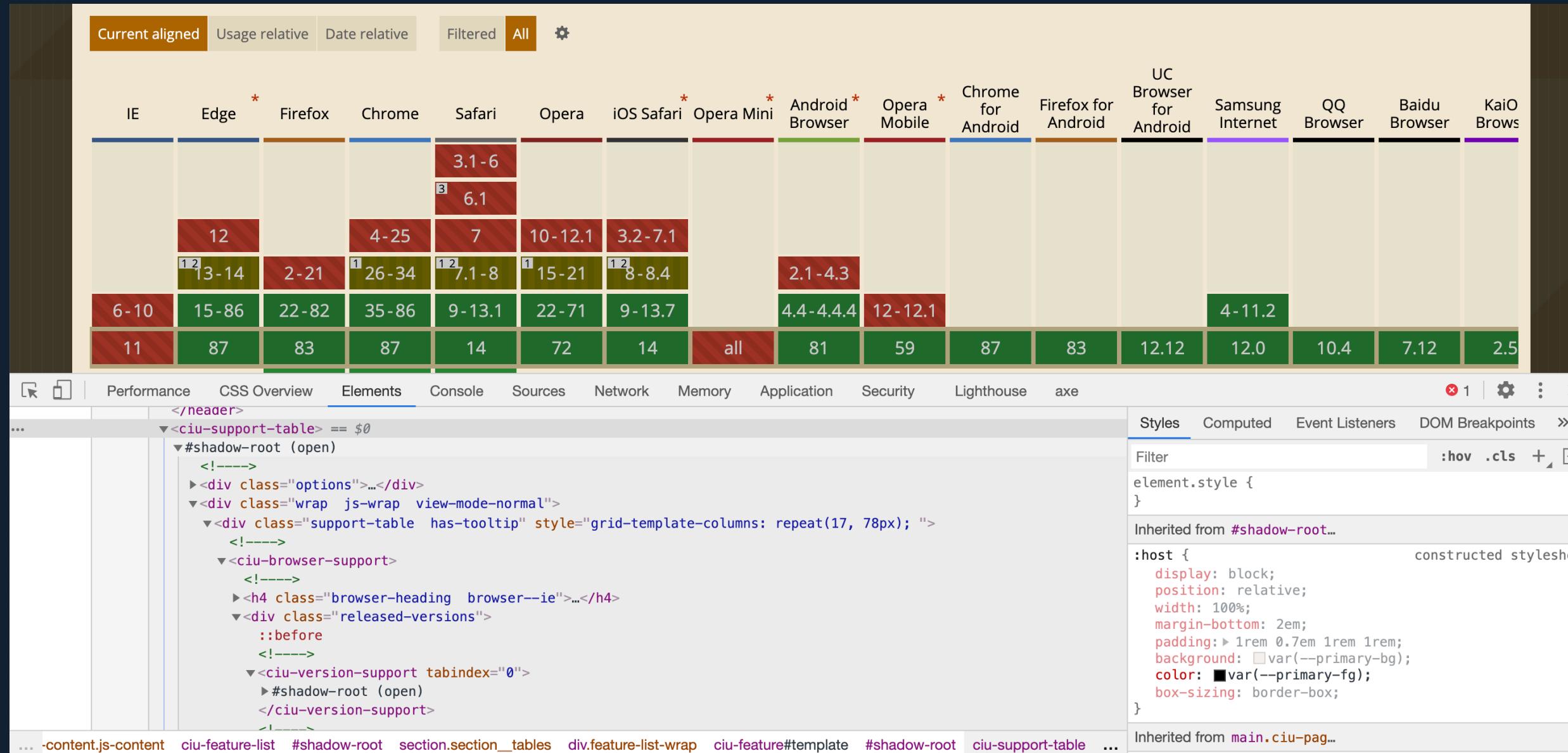
**@ILLEPIC**

# CAN I USE WEB COMPONENTS?<sup>1</sup>



<sup>1</sup> IE 11 can be supported using polyfills

# CAN CAN I USE USE WEB COMPONENTS?<sup>2</sup>



<sup>2</sup> Find many more examples at <https://wild.open-wc.org/>

# I SHOULD BUILD AN EXAMPLE WEB COMPONENT.

But what could be a relevant example in November 2020?

 2020 ELECTION

President

Senate

House

Question 1

Question 2

Mass. races

Find a race



**Joe Biden** Dem  
**264**

270 to win  
60 undecided

**Donald Trump** GOP  
**214**

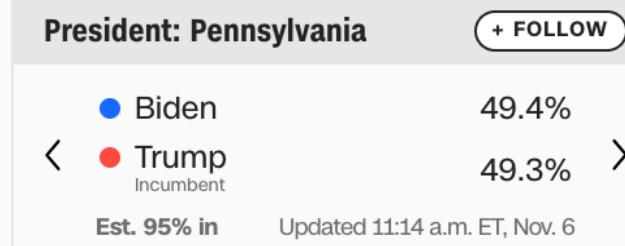


\* The Associated Press has called Joe Biden the winner in Arizona, which is reflected in this number. Some other major news organizations have yet to make a projection in the state.

**Live Election Updates**

Trump says he will ‘never give up fighting’

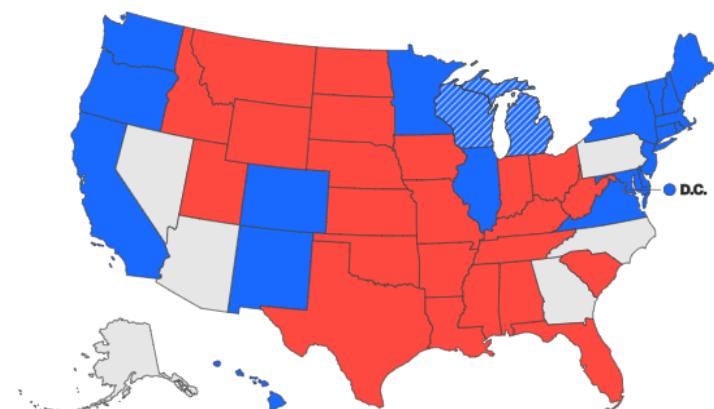
- 1:55 p.m.  
Trump vows to ‘never give up fighting,’ in new statement
- 12:47 p.m.



# Biden



## CNN PRESIDENTIAL PROJECTIONS



Updated 11:23 a.m. ET, Nov. 6



President: Pennsylvania

+ FOLLOW

● Biden  
< ● Trump  
Incumbent

49.4%  
49.3% >

Est. 95% in

Updated 11:14 a.m. ET, Nov. 6

President: Pennsylvania

+ FOLLOW

● Biden  
< ● Trump  
Incumbent

49.4%  
49.3% >

Est. 95% in

Updated 11:14 a.m. ET, Nov. 6

# Biden



THE ROAD TO 270

253

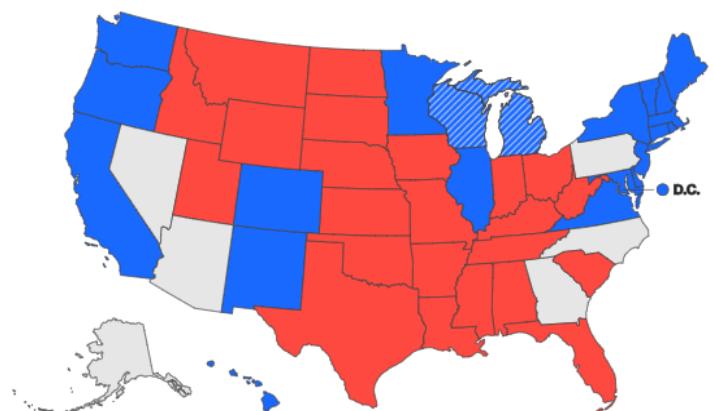
270

213



Updated 11:23 a.m. ET, Nov. 6

## CNN PRESIDENTIAL PROJECTIONS



Updated 11:23 a.m. ET, Nov. 6

BIDEN

Win  
Flip

TRUMP

Win  
Flip

# Biden



THE ROAD TO 270

253

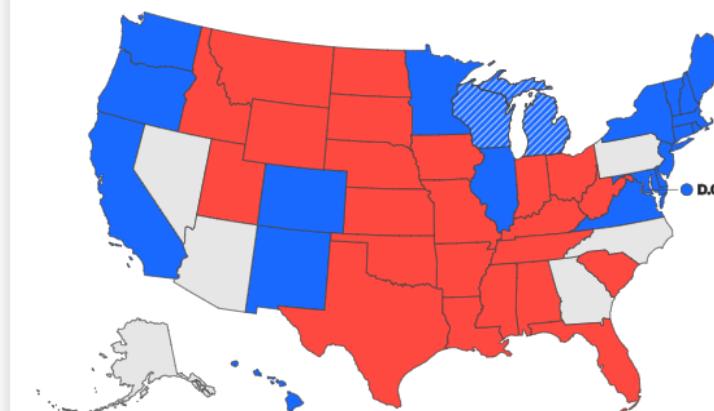
270

213



Updated 11:23 a.m. ET, Nov. 6

## CNN PRESIDENTIAL PROJECTIONS



Updated 11:23 a.m. ET, Nov. 6

BIDEN

Win  
Flip

TRUMP

Win  
Flip

47

SENATE CONTROL

47

47

SENATE CONTROL

47

President: Pennsylvania

+ FOLLOW

● Biden  
< ● Trump  
Incumbent

49.4%  
49.3% >

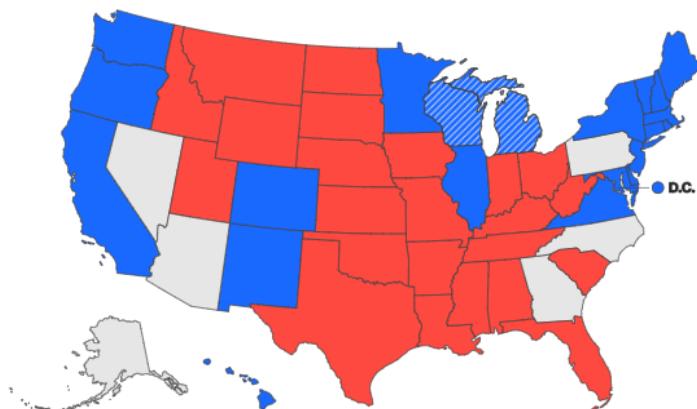
Est. 95% in

Updated 11:14 a.m. ET, Nov. 6

# Biden



## CNN PRESIDENTIAL PROJECTIONS



BIDEN  
Win Flip

TRUMP  
Win Flip

President: Pennsylvania

+ FOLLOW

● Biden  
< ● Trump  
Incumbent

49.4%  
49.3% >

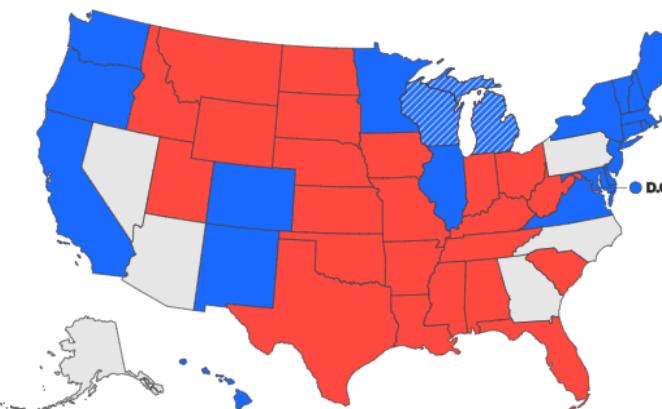
Est. 95% in

Updated 11:14 a.m. ET, Nov. 6

# Biden



## CNN PRESIDENTIAL PROJECTIONS



BIDEN  
Win Flip

TRUMP  
Win Flip

President: Pennsylvania

+ FOLLOW

● Biden  
< ● Trump  
Incumbent

49.4%  
49.3% >

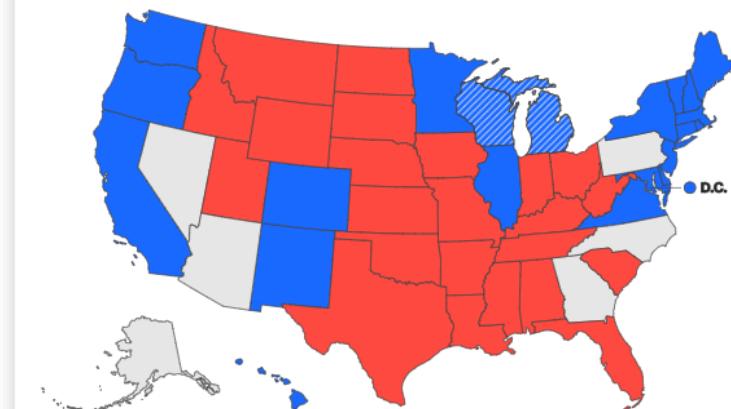
Est. 95% in

Updated 11:14 a.m. ET, Nov. 6

# Biden



## CNN PRESIDENTIAL PROJECTIONS



BIDEN  
Win Flip

TRUMP  
Win Flip

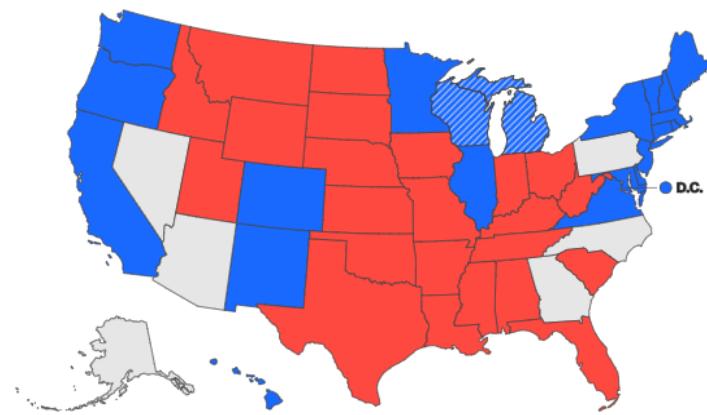
**President: Pennsylvania**

+ FOLLOW

● Biden 49.4%  
● Trump 49.3% >  
Incumbent

Est. 95% in Updated 11:14 a.m. ET, Nov. 6

# Biden

Updated 11:23 a.m. ET, Nov. 6**CNN PRESIDENTIAL PROJECTIONS**Updated 11:23 a.m. ET, Nov. 6

**BIDEN** **TRUMP**  
  
Win Flip      
Win Flip

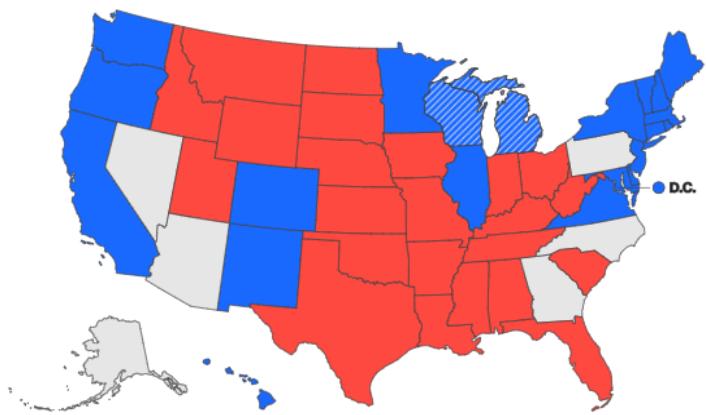
**47****SENATE CONTROL****47****President: Pennsylvania**

+ FOLLOW

● Biden 49.4%  
● Trump 49.3% >  
Incumbent

Est. 95% in Updated 11:14 a.m. ET, Nov. 6

# Biden

Updated 11:23 a.m. ET, Nov. 6**CNN PRESIDENTIAL PROJECTIONS**Updated 11:23 a.m. ET, Nov. 6

**BIDEN** **TRUMP**  
  
Win Flip      
Win Flip

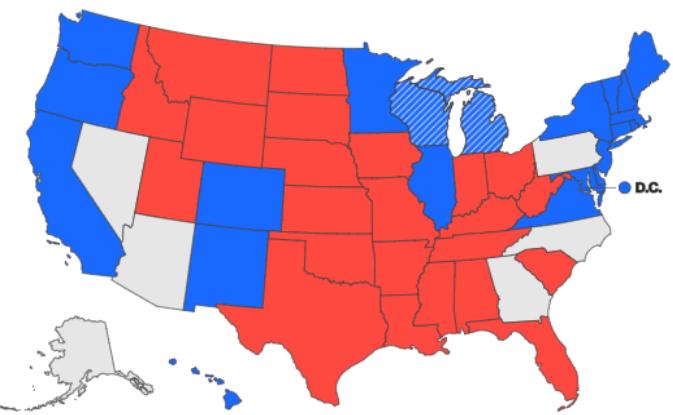
**47****SENATE CONTROL****47****President: Pennsylvania**

+ FOLLOW

● Biden 49.4%  
● Trump 49.3% >  
Incumbent

Est. 95% in Updated 11:14 a.m. ET, Nov. 6

# Biden

Updated 11:23 a.m. ET, Nov. 6**CNN PRESIDENTIAL PROJECTIONS**Updated 11:23 a.m. ET, Nov. 6

**BIDEN** **TRUMP**  
  
Win Flip      
Win Flip

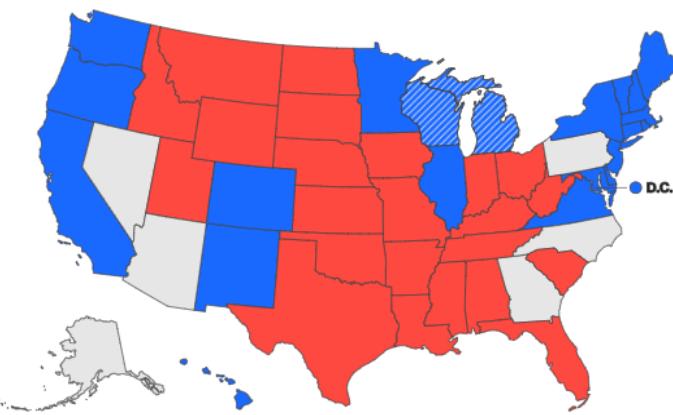
**47****SENATE CONTROL****47****President: Pennsylvania**

+ FOLLOW

● Biden 49.4%  
● Trump 49.3% >  
Incumbent

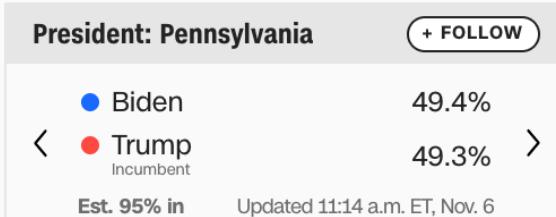
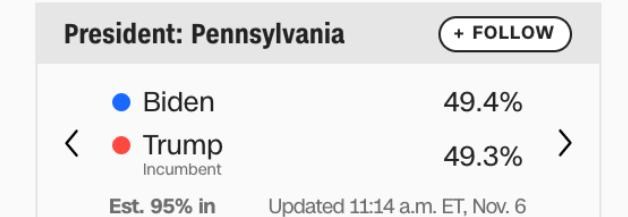
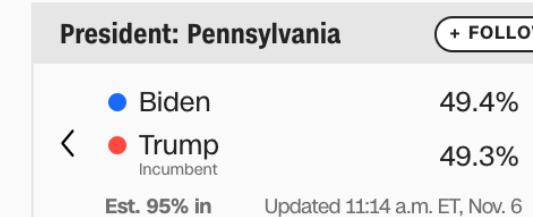
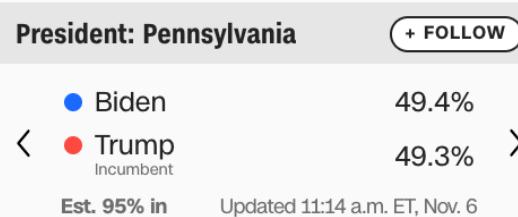
Est. 95% in Updated 11:14 a.m. ET, Nov. 6

# Biden

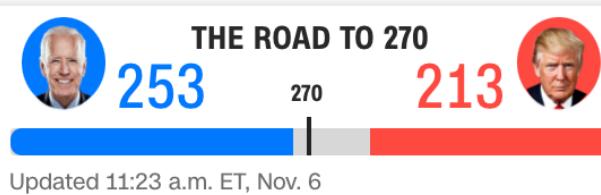
Updated 11:23 a.m. ET, Nov. 6**CNN PRESIDENTIAL PROJECTIONS**Updated 11:23 a.m. ET, Nov. 6

**BIDEN** **TRUMP**  
  
Win Flip      
Win Flip

**47****SENATE CONTROL****47**



# Biden Biden Biden Biden Biden



Updated 11:23 a.m. ET, Nov. 6



Updated 11:23 a.m. ET, Nov. 6



Updated 11:23 a.m. ET, Nov. 6

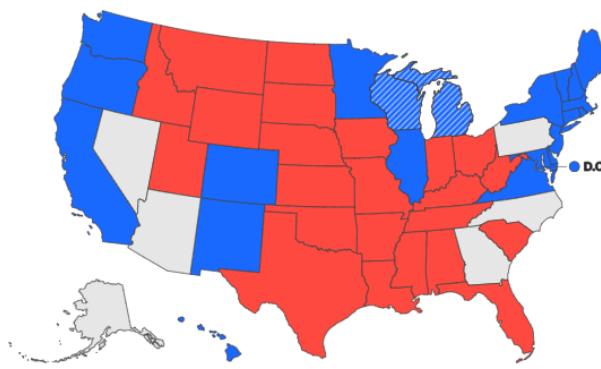


Updated 11:23 a.m. ET, Nov. 6



Updated 11:23 a.m. ET, Nov. 6

## CNN PRESIDENTIAL PROJECTIONS



Updated 11:23 a.m. ET, Nov. 6

**BIDEN**  
Win Flip

**TRUMP**  
Win Flip



Updated 11:23 a.m. ET, Nov. 6

**BIDEN**  
Win Flip

**TRUMP**  
Win Flip



Updated 11:23 a.m. ET, Nov. 6

**BIDEN**  
Win Flip

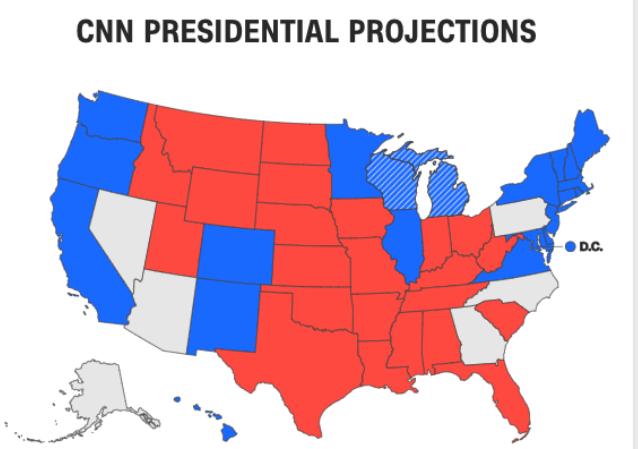
**TRUMP**  
Win Flip



Updated 11:23 a.m. ET, Nov. 6

**BIDEN**  
Win Flip

**TRUMP**  
Win Flip



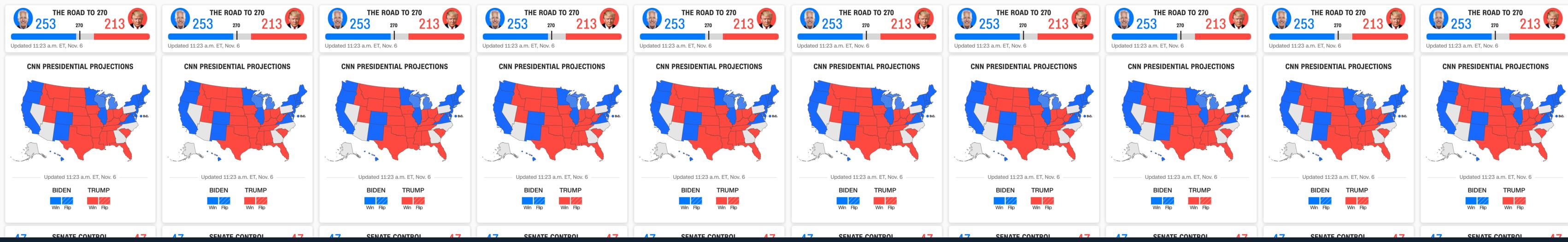
Updated 11:23 a.m. ET, Nov. 6

**BIDEN**  
Win Flip

**TRUMP**  
Win Flip

47 SENATE CONTROL

Biden Biden



Friday, November 6, 2020  
Today's Paper

62°F 74° 52°  
S&P 500 -0.14% ↓

# The New York Times

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

## Biden Leads in Pennsylvania; Win There Would Be Decisive

### State Has Electoral Votes to Put Him Over the Top

Live Updates >



President >

253

270  
TO WIN

214

Joseph R. Biden Jr.

73,879,622 votes (50.5%)

Donald J. Trump

69,772,905 votes (47.7%)

 **Dave Philipps, in Las Vegas** 39m ago  
Good morning from Las Vegas. A protest at the election center last night was peaceful — no arrests. We're expecting about 50,000 votes to be released at 9 a.m. here (noon ET). [See Nevada results](#) >

Senate >

48

50

48

DEM. +1 seat

-1 seat REP.

House >

209

218

193

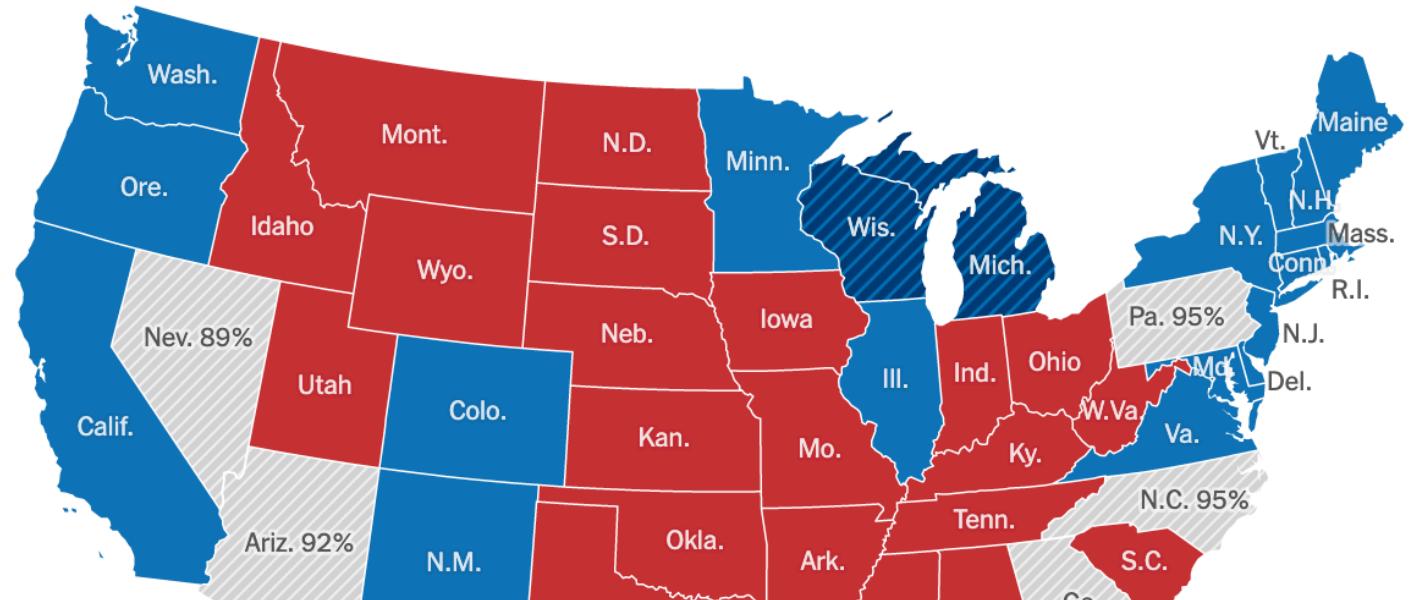
DEM. -5 seats

+6 seats REP.

## Biden Also Edges Ahead in Georgia, Gathering Strength

- Joe Biden took the lead over President Trump in Pennsylvania, whose 20 electoral votes would put him past the threshold for victory.
- Mr. Biden has steadily erased Mr. Trump's early lead in the state. He also inched up in Georgia, but his lead is diminished in Arizona.
- At the White House, Mr. Trump made a brief statement filled with falsehoods about the election process. Here's the latest.

President Senate House Photos



LIVE

• In photos: All eyes on Philadelphia. [15m ago](#)

• Here are some of the things we're watching for today as vote counting continues. [43m ago](#)

• If Trump does not concede, Biden campaign says U.S. government is 'perfectly capable of escorting trespassers out of the White House.' [1h ago](#)

• In Wilmington, Biden's supporters hope today will bring a victory celebration. [1h ago](#)

Key States >

# MY OWN PERSONAL ELECTION TRACKER

Storybook

Canvas Docs

Find components /

EXAMPLE

Introduction

COMPONENTS

Results Tracker

Primary

No Politics

## Race Between Old Men Too Close To Call

Nation tired of hitting reload in their web browsers.

**President >**

**253** **214**

270 TO WIN

**Joseph R. Biden Jr.**  
73,879,622 votes (51%)

**Donald J. Trump**  
69,772,905 votes (49%)

Controls (5) Actions

Name	Control
headline	Race Between Old Men Too Close To Call
subheadline	{"strings": ["<p>\n    Nation tired of hitting reload in their web browsers.\n  </p>"], "values": [], "type": "html", "processor": []}
race	President
total	538
candidates	[{"name": "Joseph R. Biden Jr.", "primary": 253, "secondary": 73879622, "color": "#1375b7"}, {"name": "Donald J. Trump", "primary": 214, "secondary": 69772905, "color": "#c93135"}]

# USING <ELECTION-TRACKER>

- » Import script as JS module. Could be:
- » local file
- » NPM dependency (@backlineint/results-tracker)
- » Use your custom element in markup
- » Pass data in using attributes

Note: If your component has external dependencies, you'll need to use a bundler (Webpack, Rollup, Parcel, etc.)

# CUSTOM ELEMENTS<sup>3</sup>

The screenshot shows a CodePen interface with the title "Election Results Tracker" and author "Brian Perry". The top bar includes icons for heart, save, settings, change view, and a user profile. The left sidebar shows the "HTML" tab selected, displaying the following code:

```
1 <head>
2   <script type="module" src="https://unpkg.com/@backlineint/results-tracker@1.0.0/results-tracker.js?module"></script>
3   <link rel="preconnect" href="https://fonts.gstatic.com">
4   <link href="https://fonts.googleapis.com/css2?family=Domine:wght@700&family=Libre+Franklin:wght@500;600;700&display=swap" rel="stylesheet">
5 </head>
6 <body>
7   <results-tracker
8     headline="Race Between Old Men Too Close To Call"
9     race="President"
10    total="538"
11    candidates='[
12      {"name": "Joseph R. Biden Jr.", "primary": 253, "secondary": 73879622, "color": "#1375b7"},
13      {"name": "Donald J. Trump", "primary": 214, "secondary": 69772905, "color": "#c93135"}
14    ]'
15  >
16    <p>Nation tired of hitting reload in their web browsers.</p>
17  </results-tracker>
18 </body>
```

The preview window below displays the heading "Race Between Old Men Too Close To Call" and the subtext "Nation tired of hitting reload in their web browsers.". It features a progress bar for the "President" race between Joseph R. Biden Jr. and Donald J. Trump. The bar shows Biden at 253 and Trump at 214, with 270 needed to win. The data for each candidate is also listed below the bar.

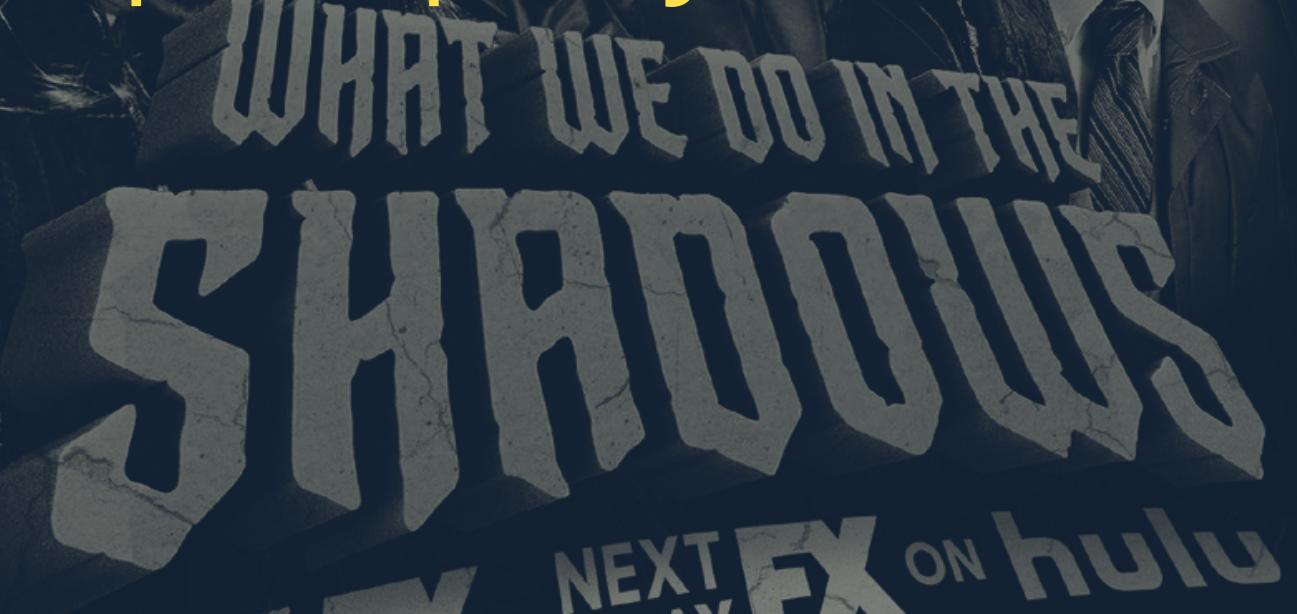
Candidate	Primary Votes	Total Votes	Color
Joseph R. Biden Jr.	253	73,879,622	#1375b7
Donald J. Trump	214	69,772,905	#c93135

At the bottom, there are navigation links for Console, Assets, Comments, and Keys, along with a timestamp "Last saved LESS THAN A MINUTE ago" and sharing options.

<sup>3</sup> <https://codepen.io/brianperry/pen/RwGPLBx>

# SHADOW DOM

- » Encapsulated DOM Tree
- » Separate from main DOM
- » Elements won't collide
- » Scoped styles
- » Super spooky



```
<!DOCTYPE html>
<html>
  ▶ <head>...</head>
  ▼ <body>
    ▼ <results-tracker headline="Race Between Old Men Too Close To Call" race="President" total="538" candidates="[
      {"name": "Joseph R. Biden Jr.", "primary": 253, "secondary": 73879622, "color": "#1375b7"}, {"name": "Donald J. Trump", "primary": 214, "secondary": 69772905, "color": "#c93135"}]>
    ▌
    ▼ #shadow-root (open)
      <!-->
      <link href="https://fonts.googleapis.com/css2?family=Domine:wght@700&family=Libre+Franklin:wght@500;600;700&display=swap" rel="stylesheet">
    ▼ <div class="results-tracker">
      ▼ <div class="results-tracker__headline">
        ▼ <h2>
          <!-->
          "Race Between Old Men Too Close To Call"
          <!-->
        </h2>
      </div>
      ▶ <div class="results-tracker__sub-headline">...</div> == $0
      ▶ <div class="results-tracker__race">...</div>
      ▶ <div class="results-tracker__counts">...</div>
      ▶ <div class="results-tracker__bars">...</div>
      ▶ <div class="results-tracker__candidates">...</div>
    </div>
    <!-->
    <p>Nation tired of hitting reload in their web browsers.</p>
  </results-tracker>
  ▶ <script>...</script>
</body>
</html>
```

# SCOPED STYLES<sup>4</sup>

The screenshot shows a code editor with three tabs: index.html, results-tracker.js, and styles.css. The styles.css tab is active, displaying the following CSS code:

```
1  /*  
2   As a general rule: No styles go into the shadow dom,  
3   and no styles get out.  
4  */  
5  
6  .results-tracker__headline {  
7    padding: 1rem;  
8    margin: 1rem 0;  
9    background: #rebeccapurple;  
10   font-size: 2rem;  
11   color: white;  
12 }  
13  
14 /* But there are exceptions... */  
15  
16 /* Inherited properties cascade */  
17 /* body {  
18   font-family: "Comic Sans MS", "Comic Sans", cursive;  
19   color: red;  
20 } */  
21
```

The browser preview shows a headline "Race Between Old Men Too Close To Call" with a subtitle "Nation tired of hitting reload in their web browsers." Below this is a chart titled "President" comparing two candidates:

Candidate	Current Votes	Needed to Win	Total Votes
Joseph R. Biden Jr.	253	270	73,879,622 votes (51%)
Donald J. Trump	214	270	69,772,905 votes (49%)

A purple callout box highlights the headline element with the text: "This div also has the class .results-tracker\_\_headline".

<sup>4</sup> <https://codesandbox.io/s/election-results-tracker-global-styling-options-w0i3e?file=/src/styles.css>

# HERE'S HOW I'VE BEEN MAKING SENSE OF THIS...

- » Only inherited properties pierce the shadow DOM
- » Everything else requires the component to expose a styling hook:
  - » CSS custom properties (variables)
  - » Classes
  - » Shadow Parts
  - » Slots

**IN CASE OF EMERGENCY ~~BREAK GLASS~~  
OPT OUT OF THE SHADOW DOM 😞**

# BUILDING <ELECTION-TRACKER> TAKE 1: VANILLA JS

# Rendering a Headline

## index.html

```
<html>
  <head>
    <title>Results Tracker Heading</title>
    <meta charset="UTF-8" />
  </head>

  <body>
    <results-tracker headline="Race Between Old Men Too Close To Call" />

    <script type="module" src="src/results-tracker.js"></script>
  </body>
</html>
```

# results-tracker.js

```
class ResultsTracker extends HTMLElement {
  constructor() {
    // Always call super first in constructor
    super();

    // Create a shadow root
    const shadow = this.attachShadow({ mode: "open" }); // sets and returns 'this.shadowRoot'

    // Create wrapping element
    const wrapper = document.createElement("div");
    wrapper.setAttribute("class", "results-tracker");

    this.headlineElement = document.createElement("h2");
    this.headlineElement.setAttribute("class", "results-tracker__headline");
    this.headlineElement.textContent = this.getAttribute("headline");
    wrapper.appendChild(this.headlineElement);

    // Attach the results tracker to the shadow DOM.
    shadow.appendChild(wrapper);
  }
}

// Define the new element
customElements.define("results-tracker", ResultsTracker);
```

# Refactoring to use <template>

```
class ResultsTracker extends HTMLElement {
  constructor() {
    super();
    this.shadow = this.attachShadow({ mode: "open" });

    // Templates are not referenced in the DOM, but can be referenced / cloned using js
    const template = document.createElement("template");

    template.innerHTML = `
      <div class="results-tracker">
        <div class="results-tracker__headline">
          <h2>${this.getAttribute("headline")}</h2>
        </div>
      </div>
    `;

    // Attach the template to the Shadow DOM
    this.shadow.appendChild(template.content);
  }
}

customElements.define("results-tracker", ResultsTracker);
```

# Add scoped styling

```
class ResultsTracker extends HTMLElement {
  constructor() {
    super();
    /* Removed for brevity... */

    // Create CSS to apply to the shadow dom
    const style = document.createElement("style");

    style.textContent = `
      :host {
        font-family: 'Libre Franklin', helvetica, arial, sans-serif;
      }
      h2 {
        margin: .5rem 0;
        font-family: 'Domine', serif;
        font-weight: 700;
        font-size: 36px;
        text-align: center;
      }
    `;

    // Attach the styles to the shadow dom
    shadow.appendChild(style);

    /* Removed for brevity... */
  }
}
```

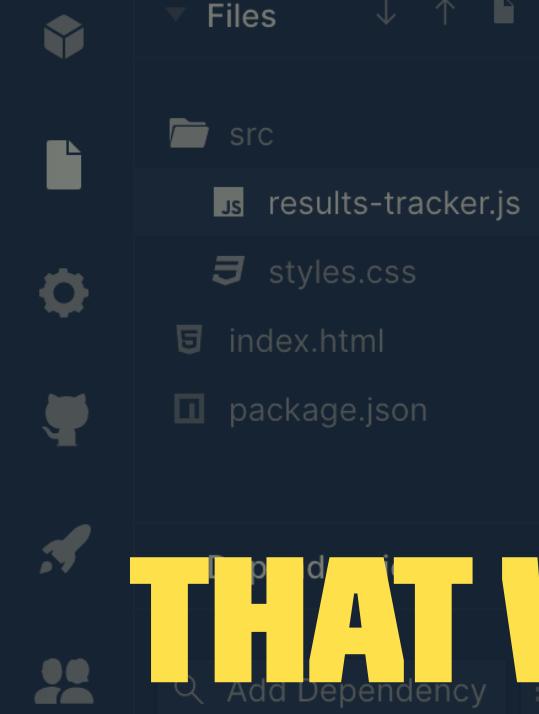
# Observe attributes and re-render if changed

```
class ResultsTracker extends HTMLElement {
  // Specify observed attributes for attributeChangedCallback
  static get observedAttributes() {
    return ["headline"];
  }

  constructor() {
    /* Removed for brevity */
  }

  // Custom element lifecycle callback function
  attributeChangedCallback(name, oldValue, newValue) {
    // Compare old to new to prevent unnecessary re-rendering
    if (oldValue !== newValue && name === "headline") {
      this.shadow.querySelector(
        ".results-tracker__headline h2"
      ).textContent = newValue;
    }
  }
}

// Define the new element
customElements.define("results-tracker", ResultsTracker);
```



Race Between Old Men Too Close To Call

# THAT WAS TOO MUCH WORK FOR A HEADLINE...<sup>5</sup>

# SHOULDN'T THIS BE EASIER?

```
1 class ResultsTracker extends HTMLElement {  
2     // Specify observed attributes so that attributeChangedCallback  
3     static get observedAttributes() {  
4         return ["headline"];  
5     }  
6  
7     constructor() {  
8         // Always call super first in constructor  
9         super();  
10        this.shadow = this.attachShadow({ mode: "open" }); // sets up the shadow DOM  
11        const template = document.createElement("template");  
12        template.innerHTML = `  
13            <div class="results-tracker">  
14                <div class="results-tracker__headline">  
15                    <h2></h2>  
16                </div>  
17            </div>  
18        `;  
19        // Attach the template to the Shadow DOM  
20        this.shadow.appendChild(template.content);  
21        // Create CSS to apply to the shadow dom  
22        const style = document.createElement("style");  
23        style.innerHTML = ".results-tracker { display: flex; align-items: center; justify-content: center; } .results-tracker__headline { font-size: 2em; font-weight: bold; }";  
24        this.shadow.appendChild(style);  
25    }  
26  
27    // Attach the template to the Shadow DOM  
28    this.shadow.appendChild(template.content);  
29  
30    const style = document.createElement("style");  
31    style.innerHTML = ".results-tracker { display: flex; align-items: center; justify-content: center; } .results-tracker__headline { font-size: 2em; font-weight: bold; }";  
32    this.shadow.appendChild(style);  
33}
```

<sup>5</sup> <https://codesandbox.io/s/results-tracker-heading-vanilla-js-lunes>

**WAIT FOR IT**

The image features a bold, sans-serif font where the letters 'I' and 'T' are merged into a single vertical stroke. To the right of the word 'IT', there are three small, yellow, rounded circles of decreasing size.

**SO.  
MANY.  
LIBRARIES.**

- » LitElement
- » Stencil
- » FastElement
- » Haunted
- » Hybrids
- » Many more...

**AREN'T WE JUST BACK WHERE WE  
STARTED?**

Only one way to find out...

# BUILDING <ELECTION-TRACKER> TAKE 2: LIT-ELEMENT

# Rendering a headline (LitElement version)

## index.html

```
<html>
  <head>
    <title>Results Tracker Heading - Lit</title>
    <meta charset="UTF-8" />
  </head>

  <body>
    <results-tracker headline="Race Between Old Men Too Close To Call" />

    <script type="module" src="src/results-tracker.js"></script>
  </body>
</html>
```

(same as vanilla js version)

# results-tracker.js

```
import { LitElement, html } from "lit-element";

export class ResultsTracker extends LitElement {
  static get properties() {
    return {
      headline: { type: String }
    };
  }

  render() {
    return html`
      <div class="results-tracker">
        <div class="results-tracker__headline">
          <h2>${this.headline}</h2>
        </div>
      </div>
    `;
  }
}

window.customElements.define("results-tracker", ResultsTracker);
```

# A LOT WITH LESS

- » Renders custom element
- » Templating
- » Observes updates to attributes

Just need to add scoped styles.

# Add scoped styling

```
import { LitElement, html, css } from "lit-element";

export class ResultsTracker extends LitElement {
  static get styles() {
    return css`
      :host {
        font-family: "Libre Franklin", helvetica, arial, sans-serif;
      }
      .results-tracker__headline h2 {
        margin: 0.5rem 0;
        font-family: "Domine", serif;
        font-weight: 700;
        font-size: 36px;
        text-align: center;
      }
    `;
  }

  // Properties...

  // Render method...
}

window.customElements.define("results-tracker", ResultsTracker);
```

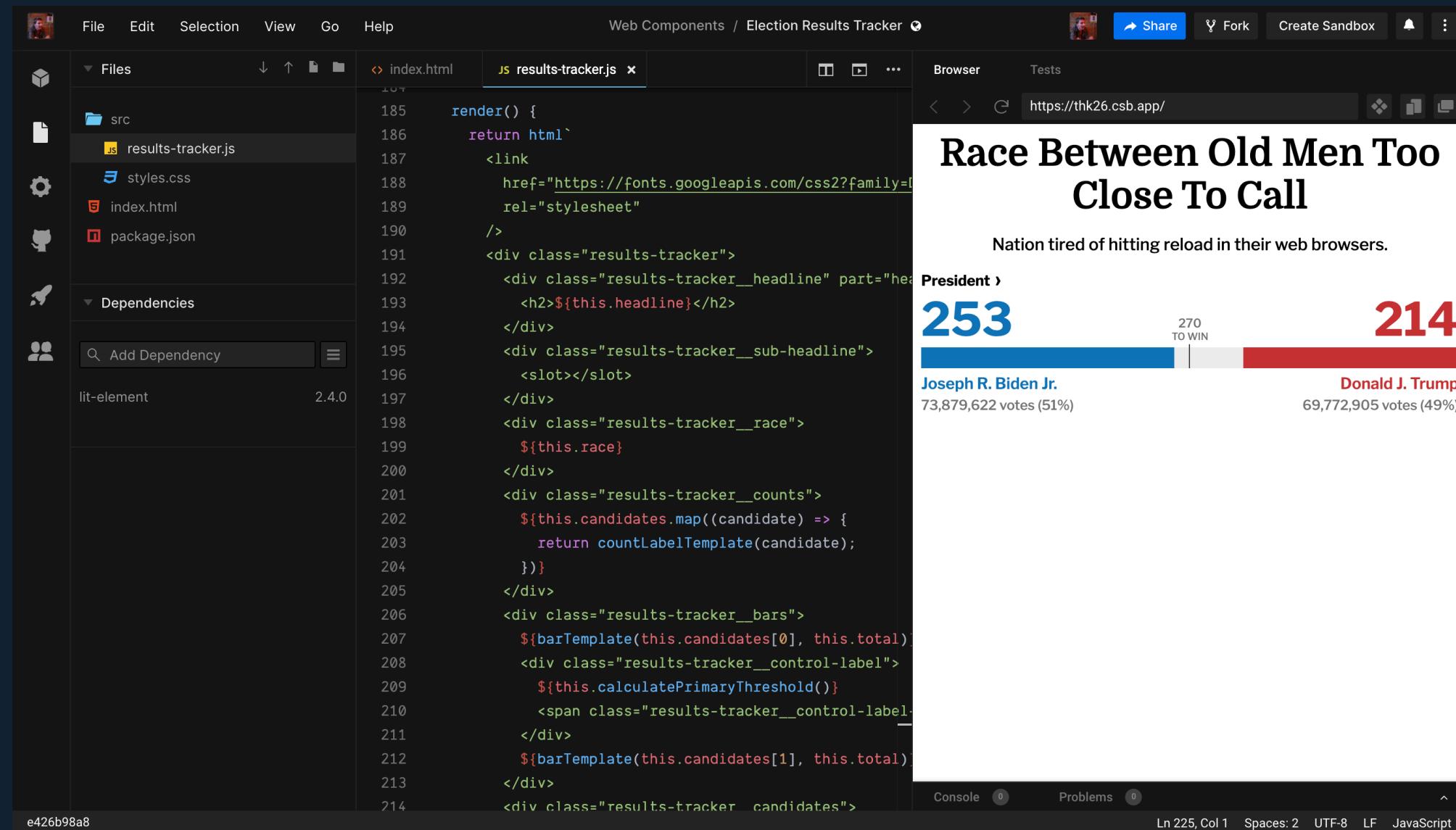
# Converting attributes

```
// Vanilla JS - have to manually transform string attributes
processCandidates() {
  this.candidates = JSON.parse(this.getAttribute('candidates'));
  // ...
}

// Define the type of your property, and LitElement will automatically
// handle conversion for you.
static get properties() {
  return {
    /**
     * An array of objects containing data for each candidate
     */
    candidates: {type: Array},
  };
}
```

(And many other DX niceties)

# Feels like a more appropriate amount of work for a headline... and especially the full results-tracker<sup>6</sup>



The screenshot shows a web component editor interface. On the left, the file structure is visible with files like index.html, results-tracker.js, styles.css, and package.json. The main area displays the code for results-tracker.js, which includes rendering HTML, linking fonts, and displaying election results for President Biden and Trump. The preview window on the right shows the results: Biden has 253 electoral votes and Trump has 214, with 270 needed to win. The nation is described as being "tired of hitting reload in their web browsers".

```
185 render() {
186   return html` 
187     <link
188       href="https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@400;700&display=swap"
189       rel="stylesheet"
190     />
191     <div class="results-tracker">
192       <div class="results-tracker__headline" part="headline">
193         <h2>${this.headline}</h2>
194       </div>
195       <div class="results-tracker__sub-headline">
196         <slot></slot>
197       </div>
198       <div class="results-tracker__race">
199         ${this.race}
200       </div>
201       <div class="results-tracker__counts">
202         ${this.candidates.map((candidate) => {
203           return countLabelTemplate(candidate);
204         })}
205       </div>
206       <div class="results-tracker__bars">
207         ${barTemplate(this.candidates[0], this.total)}
208         <div class="results-tracker__control-label">
209           ${this.calculatePrimaryThreshold()}
210           <span class="results-tracker__control-label">
211             ${barTemplate(this.candidates[1], this.total)}
212           </div>
213         </div>
214         <div class="results-tracker__candidates">
```

Race Between Old Men Too Close To Call  
Nation tired of hitting reload in their web browsers.

President >

Candidate	Votes	Percentage
Joseph R. Biden Jr.	73,879,622	(51%)
Donald J. Trump	69,772,905	(49%)

253      214  
270 TO WIN

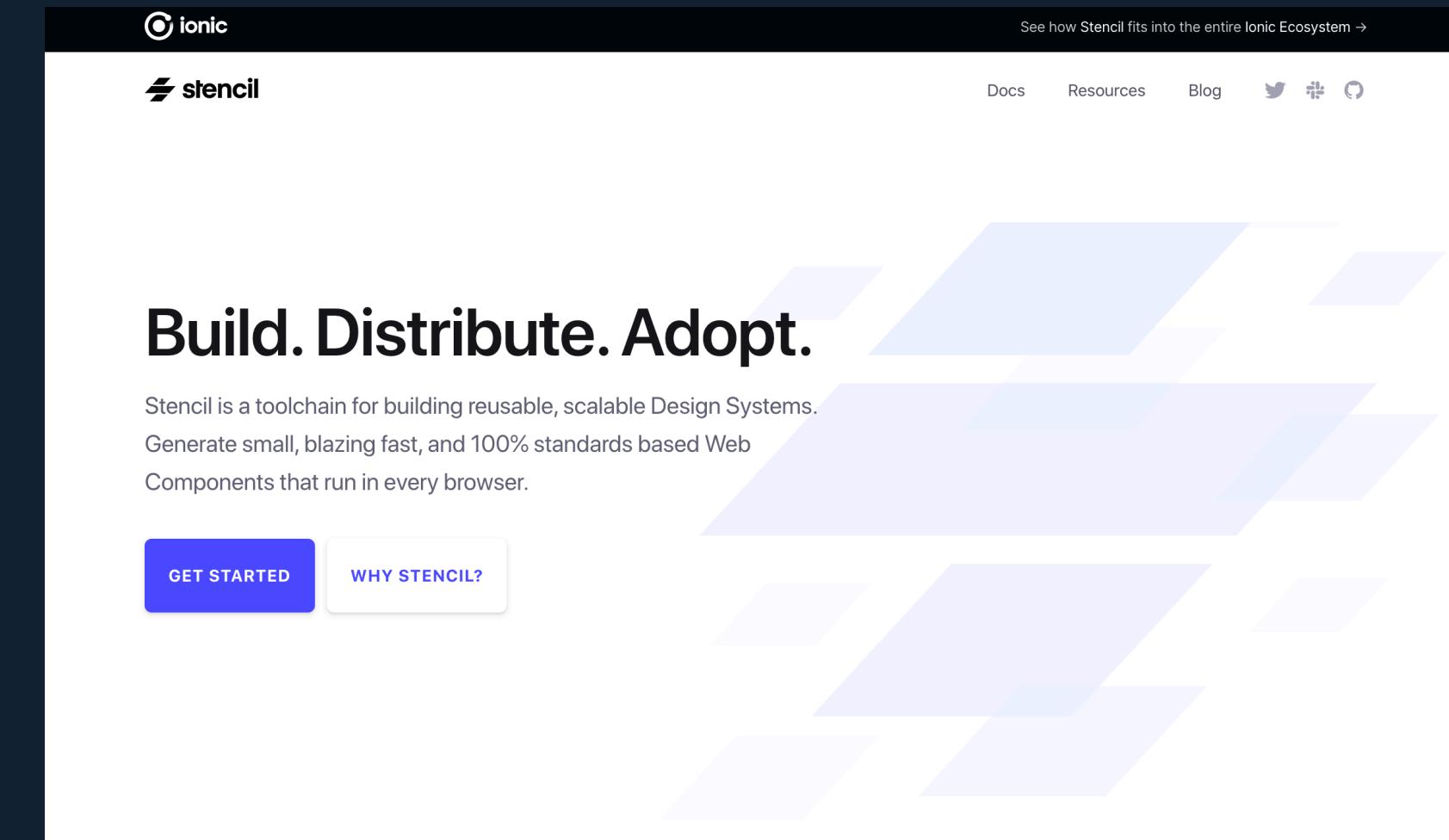
<sup>6</sup> <https://codesandbox.io/s/election-results-tracker-thk26>

# STENCIL

“Compiler that generates web components”

Provides extra capabilities on top of Web Components:

- » Prerendering
- » Objects-as-properties
- » Virtual DOM
- » JSX
- » Async Rendering



The screenshot shows the Stencil landing page. At the top, there's a header with the Ionic logo and a link to see how Stencil fits into the entire Ionic Ecosystem. Below the header, the Stencil logo is displayed. The main headline reads "Build. Distribute. Adopt." followed by a description: "Stencil is a toolchain for building reusable, scalable Design Systems. Generate small, blazing fast, and 100% standards based Web Components that run in every browser." At the bottom, there are two buttons: a blue "GET STARTED" button and a white "WHY STENCIL?" button.

# VUE

Supports web components as a build target.

But...

It still requires the Vue library as a global dependency.



Vue CLI

Guide Config Reference Plugins Migrating From v3 Changelog Languages GitHub

**Basics**

- Instant Prototyping
- Creating a Project
- Plugins and Presets
- CLI Service

**Development**

- Browser Compatibility
- HTML and Static Assets
- Working with CSS
- Working with Webpack
- Modes and Environment Variables

**Build Targets**

- App
- Library
- [Vue vs. JS/TS Entry Files](#)
- Web Component
  - Bundle that Registers Multiple Web Components
  - Async Web Component
  - Using vuex in builds
- Deployment
- Troubleshooting

**Web Component**

**Note on Compatibility**  
Web Component mode does not support IE11 and below. [More details](#)

**Note on Vue Dependency**  
In web component mode, Vue is *externalized*. This means the bundle will not bundle Vue even if your code imports Vue. The bundle will assume `Vue` is available on the host page as a global variable.  
To avoid this behavior provide `--inline-vue` flag to `build` command.

```
vue-cli-service build --target wc --inline-vue
```

You can build a single entry as a web component using

```
vue-cli-service build --target wc --name my-element [entry]
```

Note that the entry should be a `*.vue` file. Vue CLI will automatically wrap and register the component as a Web Component for you, and there's no need to do this yourself in `main.js`. You can use `main.js` as a demo app solely for development.

The build will produce a single JavaScript file (and its minified version) with everything inlined. The

# USING WEB COMPONENTS WITH A FRAMEWORK

custom-elements-  
everywhere.com outlines  
support for many frameworks.

Support is pretty solid  
across the board.

React has some notable  
limitations. 😞

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

Angular components can listen to native DOM events dispatched from Custom Elements. It supports all styles of events (lowercase, camelCase, kebab-case, etc).

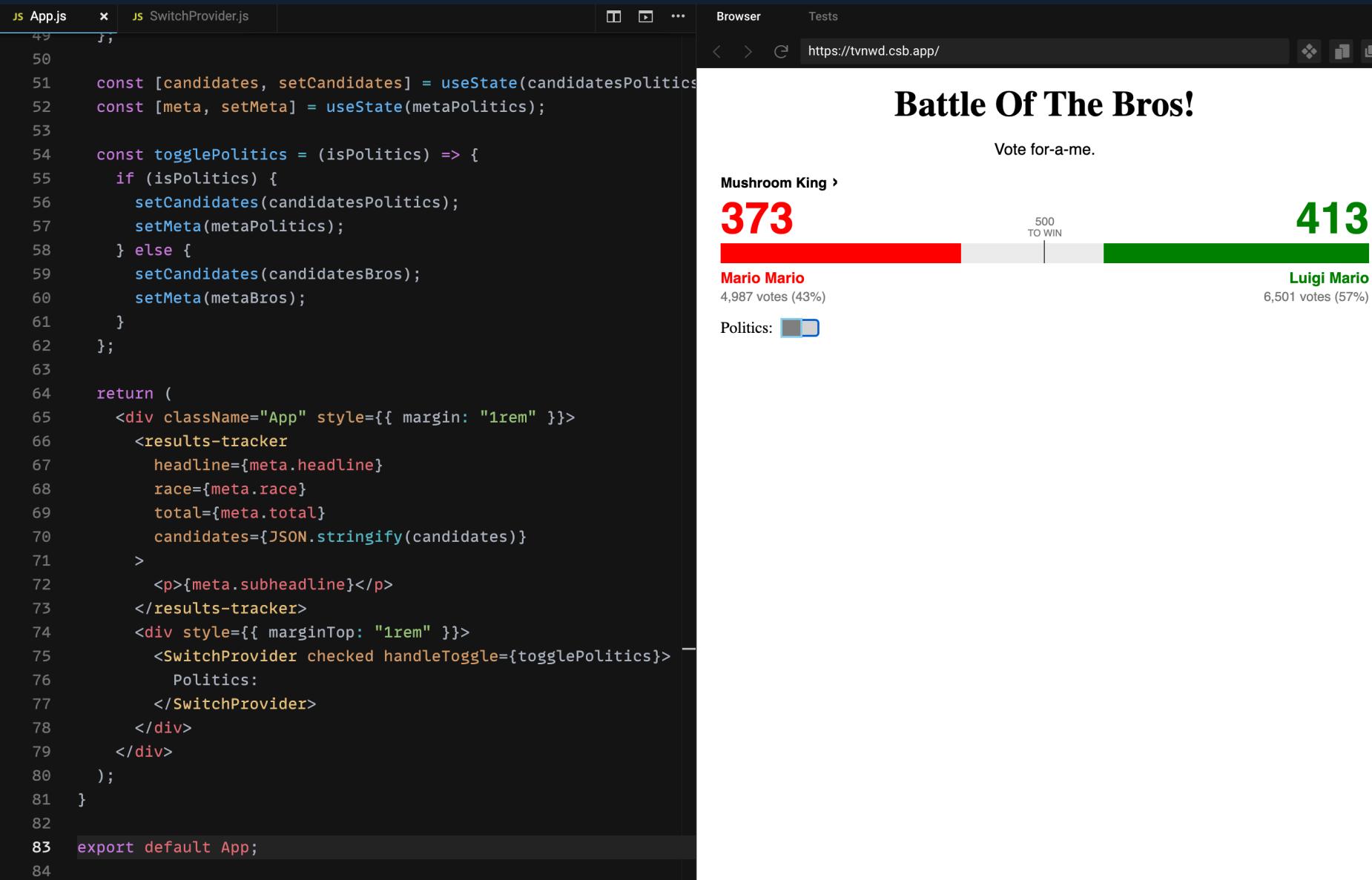
[View the tests](#)

Related Issues



Yay! No open issues!

# REACT + CUSTOM ELEMENTS<sup>7</sup>



The screenshot shows a development environment with two tabs: 'App.js' and 'SwitchProvider.js'. The 'App.js' tab displays the following code:

```
49
50
51  const [candidates, setCandidates] = useState(candidatesPolitics);
52  const [meta, setMeta] = useState(metaPolitics);
53
54  const togglePolitics = (isPolitics) => {
55    if (isPolitics) {
56      setCandidates(candidatesPolitics);
57      setMeta(metaPolitics);
58    } else {
59      setCandidates(candidatesBros);
60      setMeta(metaBros);
61    }
62  };
63
64  return (
65    <div className="App" style={{ margin: "1rem" }}>
66      <results-tracker
67        headline={meta.headline}
68        race={meta.race}
69        total={meta.total}
70        candidates={JSON.stringify(candidates)}
71      >
72        <p>{meta.subheadline}</p>
73      </results-tracker>
74      <div style={{ marginTop: "1rem" }}>
75        <SwitchProvider checked handleToggle={togglePolitics}> —
76          Politics:
77        </SwitchProvider>
78      </div>
79    </div>
80  );
81}
82
83 export default App;
```

The browser window shows a web application titled 'Battle Of The Bros!' with the subtitle 'Vote for-a-me.' It features a progress bar comparing 'Mushroom King' (373 votes) and 'Luigi Mario' (413 votes), with a goal of 500 TO WIN. Below the progress bar, it shows 'Mario Mario' (4,987 votes, 43%) and 'Luigi Mario' (6,501 votes, 57%). A 'Politics:' switch is also present.

<sup>7</sup> <https://codesandbox.io/s/web-components-with-react-tvnwd>

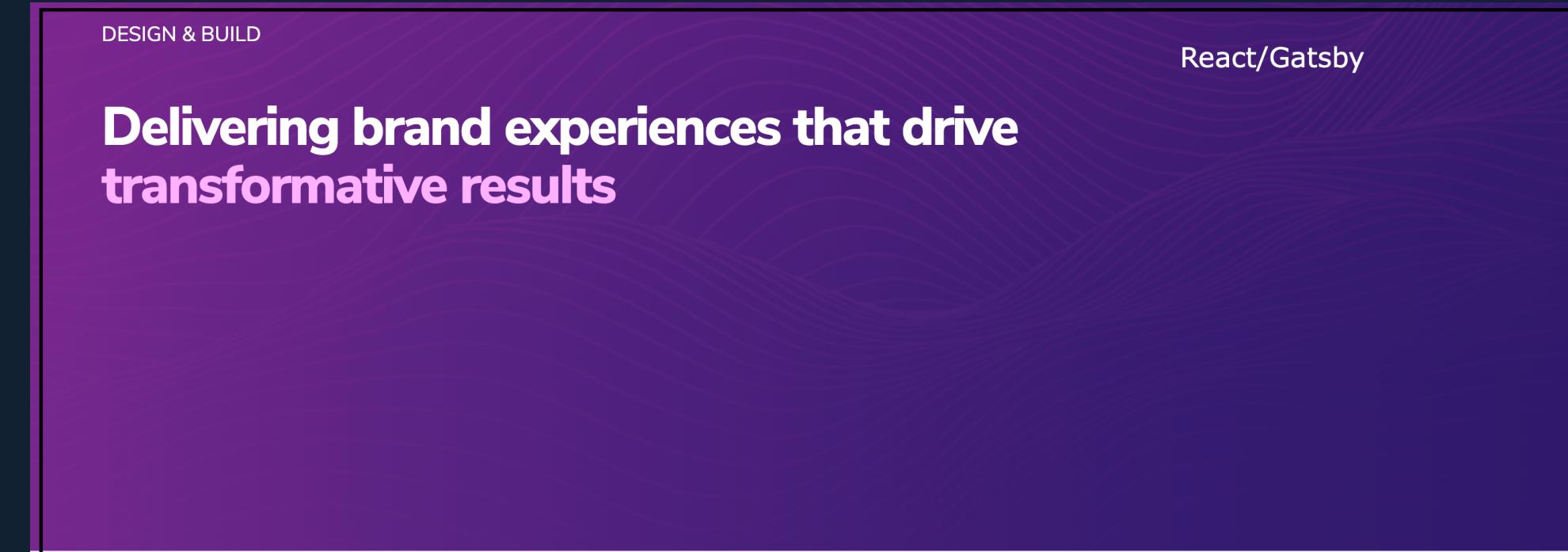
# MANAGING APPLICATION STATE

- » Didn't come across any clear pattern or best practice.
- » Could roll your own.
- » Could use any JS based framework or state management library.
- » Would be nice if a default standard existed (think React context)
- » Maybe this is an unreasonable expectation...

# MY (SLIGHTLY) MORE EDUCATED VIEWS ON WEB COMPONENTS

- » This was pretty hard to learn!
- » I'd turn to this for special purpose components today.
- » Not yet comfortable enough for a full app/design system.
- » I would use a library, but one close to the vanilla API.

# A (possibly unusual?) use case



DESIGN & BUILD

React/Gatsby

**Delivering brand experiences that drive transformative results**

Drupal Markup

**Engaging experiences are rooted in strategy and insights**

Armed with critical insights about our client's business and customers, we assemble cross-disciplinary teams of thinkers and makers to craft meaningful, results-driven experiences that authentically express the brand's purpose. Our unique co-innovation enablement model guides this process while helping clients boost their digital maturity through best-in-class talent, methods, data, and technology.

**500+**  
BRAND EXPERIENCES LAUNCHED

---

**60+**  
LANGUAGES, FRAMEWORKS & PLATFORMS

# ON AN INFINITE TIMESCALE...

- » I think some version of this concept will win out.
- » But how infinite is that timescale?
- » And will it be this take on web components?

At least I've got this cool election tracker.

**THANKS!**

brian.perry@bounteous.com

@bricomedy

brianperry.dev



# Contribution Day

Saturday, February 20, 2021  
12:00pm - 3:30pm

## Planned Workshops

- First-time contributor workshop
- Introduction to Merge Requests
- Mentored Tooling

## Planned Initiatives

- Olivero Theme
- SimplyTest
- Drupal Recipes