

Build your next App with Web Components



Hey!

Who is this guy?

Name: Andy Desmarais

Twitter/Github/Reddit: terodox

Website: terodox.tech

Background: Full stack
developer for 17 years



What's the problem?

The framework wars



“

“Every line of frontend code is tech debt when it’s written in a framework.”

- Me

We've all lost
sleep to the
framework wars

- Frameworks equate to vendor lock-in
- Upgrade fatigue is real
- So many dependencies
- You always pick wrong

A NEW ALLY HAS APPEARED



WEBCOMPONENTS

The best framework for is the web platform

*Everything you need - components,
lifecycle methods, style encapsulation,
slots, AND great documentation!*



Web Components

It's in the name!

Web components are the combination of a few pieces of web technology

Custom Elements - define your own elements that the DOM will natively render

Shadow DOM - Encapsulate both your HTML and CSS to isolate them from the rest of the DOM

Templates - Faster render times using predefined `<template>` elements

Slots - Allow your consumers to provide their own content.

Similar to frameworks:

Angular - Content Projection

React - children

Vue - Slots

Custom Elements

Bring your own elements to the DOM!

Custom Elements allow you to define your own elements that the DOM will render natively!

Minimal requirements:

- Must be all lowercase
- Must contain a hyphen “-”

Lifecycle Methods!

- `connectedCallback` - element added to the DOM
- `disconnectedCallback` - element removed from the DOM
- `attributeChangedCallback` - fired every time an observed attribute is changed

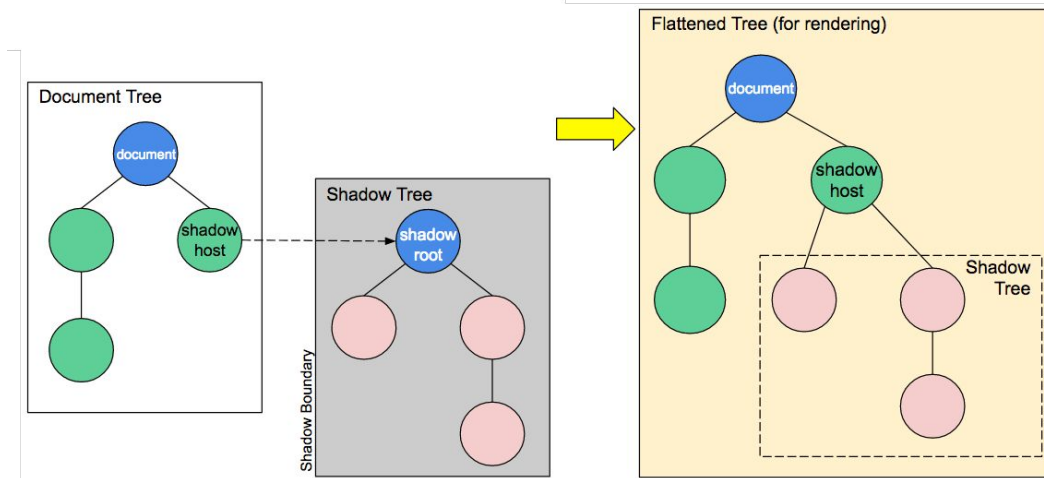
Style Encapsulation

And it doesn't even need to be obfuscated!

The shadowDom provides style encapsulation!

Native to the browser, no need for plugins!

Minimal build tooling if you want SCSS.



Slots allow us to setup components that will have at least part of their content provided by the consumer.

Slots


Allow consumers to provide the content!

Basic Slot Example

Andy Desmarais

```
HTML
1 <my-card>
2   <span slot="title">This is the
   card title</span>
3   
4   <span slot="caption">Meltwater
   Logo</span>
5 </my-card>
```

This is the card title



Meltwater Logo

Documentation

MDN has you covered!

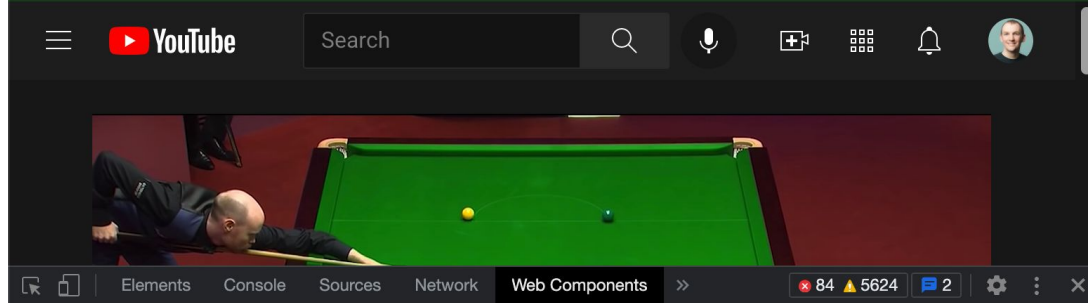
Web Components are built into all modern browser. This means out documentation is available via the best curated docs on the web: MDN!

Here are the links for everything covered so far:

- [Web Components](#)
- [Custom Elements](#)
- [Shadow DOM](#)
- [Slots](#)

Is it really enterprise ready?

Tooling, Delivery, and Support OH MY!



Filter by name

```
▼ ytd-app <ytd-app scrollbar-rework standardized-themed-scrollbar scrollbar-color>
  yt-activity-manager <yt-activity-manager class="st...
  yt-guide-manager <yt-guide-manager id="guide-se...
  yt-gfeedback-manager <yt-gfeedback-manager class="s...
  yt-mdx-manager <yt-mdx-manager class="styl...
  yt-navigation-manager <yt-navigation-manager id="nav...
  yt-network-manager <yt-network-manager id="net" c...
  yt-playlist-manager <yt-playlist-manager class="st...
  yt-player-manager <yt-player-manager player-cont...
  yt-hotkey-manager <yt-hotkey-manager class="styl...
  yt-visibility-monitor <yt-visibility-monitor id="vis...
  yt-history-manager <yt-history-manager id="histor...
  ▼ ytd-masthead <ytd-masthead id="masthead" lo...
```

ytd-app

▼ Properties

▶ **data:** ("page": "watch", "endpoint": "clickTracki...
endpointHandlerTarget: <ytd-app scrollbar-rework standardized-themed-scrollbar scrollbar-color>

errorMessage:

guidelsVisibleButNotPersistent

guidePersistentAndVisible

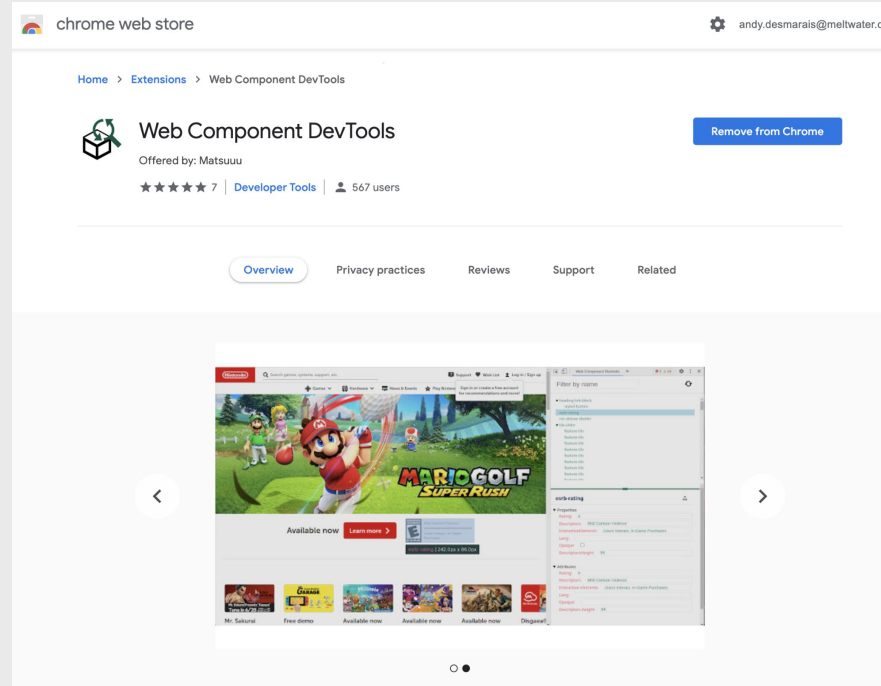
guideScrolledByTouch

hasError

Tooling

Web Component Dev Tools

No, I wasn't trying to repeat myself 🙄 The extension is called **Web Component Dev Tools!**



[Link to the extension](#)

Delivery

It's just javascript

One of the best parts of Web Components is they are just javascript. Include your script tag and use the tag in your markup!

Basic Slot Example


Andy Desmarais

```
HTML
1 <my-card>
2   <span slot="title">This is the card
  title</span>
3   
4   <span slot="caption">Meltwater
  Logo</span>
5 </my-card>
```

```
CSS
```

```
JS
1 class MyCard extends HTMLElement {
2   constructor() {
3     super();
4     this.attachShadow({ mode: 'open' });
5     this.shadowRoot.innerHTML = `
```

This is the card title



Meltwater Logo

The simplest version of any of the bundling tools can provide you with these small parts you want.

Delivery

I want minification and obfuscation

```
webpack.config.js > ...
1  module.exports = {
2    ...
3    entry: {
4      ... './my-card': './src/my-card.mjs'
5    },
6    mode: 'production',
7    optimization: {
8      ... minimize: true,
9    }
10 };
```

```
rollup.config.js > [e] default > [o] output
1  import { terser } from "rollup-plugin-terser";
2
3  export default {
4    ... input: 'src/my-card.mjs',
5    ... output: [
6      ... {
7        ... dir: "dist",
8        ... format: "esm",
9        ... exports: "named",
10     ... }
11   ],
12   ... plugins: [terser()],
13 };
```

And now you just need to drop your script tag in your html!

Delivery

Just a script tag

```
demo > index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <script type="module" src="src/my-card.mjs"></script>
9 </head>
10 <body>
11   <my-card>
12     <span slot="title">This is the card title</span>
13     
14     <span slot="caption">Meltwater Logo</span>
15   </my-card>
16 </body>
17 </html>
```


Can I use it - Yes you can!

Support

IT'S EVERYWHERE!

That you care about

Custom Elements (V1) LS Usage % of all users Global 77.04% + 18.11% = 95.15%

One of the key features of the Web Components system, custom elements allow new HTML tags to be defined.

Current aligned Usage relative Date relative Filtered All

IE	Edge	Firefox	Chrome	Safari	Opera	Safari on iOS	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
		2-49														
		20-58	4-53		10-40								4			
	12-18	59-62	54-66	3.1-10	41-63	3.2-10.2							5.4			
6-10	79-93	63-91	67-93	10.1-14.1	64-78	10.3-14.8		2.1-4.4.4	12-12.1				6.2-14.0			
11	94	92	94	15	79	15	all	94	64	94	92	12.12	15.0	10.4	7.12	2.5
		93-94	95-97	TP												

Notes Test on a real browser Known issues (0) Resources (10) Feedback

- 1 Supports "Autonomous custom elements" but not "Customized built-in elements" see [WebKit bug 182671](#).
- 2 Enabled through the `dom.webcomponents.enabled` preference in `about:config`.
- 3 Enabled through the `dom.webcomponents.customElements.enabled` preference in `about:config`.

But how do I handle...?

Let's answer some of the most common questions I get around building with Web Components

- Dynamic Templates
- Data Binding
- Routing
- Scss
- State management



Dynamic templates

- It's just javascript!
- Basic string concatenation
- If you want advanced rendering, check out [lit-html](#)! Only 6kb and allows:
 - Dynamic template rendering
 - Data binding
 - Event binding
 - Attribute binding
 - Property binding



Data binding

- Data binding can be done through properties!
- A web component is an object just like any other. You can pass complex data types to properties.
- [lit-html](#) can also assist here if we want a more fluent binding syntax

Routing

- [Vaadin Router](#)
- Features:
 - Basic router outlet routing
 - Child routes
 - Fallback routes (Eg 404)
 - Route params
 - Redirect
 - Lazy loading js bundles



Webpack and Rollup both support it!

Scss

Webpack

```
module: {
  rules: [
    {
      test: /\.scss$/,
      use: [
        'raw-loader',
        {
          loader: 'sass-loader',
          options: {
            sassOptions: {
              exclude: /node_modules/,
            }
          }
        },
      ],
    },
  ],
}
```

Rollup

```
import scss from 'rollup-plugin-scss';
```

```
... scss({
...   output: false,
... })
```

State Management

There are a lot of ways to manage state, but here are a few options:

- A singleton - Keeps shared state in one place
- Redux - Usually overkill
- Using graphql - Use Apollo for that



Thank you!

