

Zero dependency front-end components with Web Components

sthlm.js Nov 27, 2019

Web standards and vanilla JS FTW!

HTML, JS, CSS

Layout (Blink), JS (V8), APIs

Including of course **web components**

Hi, I'm Eric
Entrepreneur and
full-stack developer

I love simple, elegant code.
I hate black boxes and “magic”.

Frameworks and libraries
are great, but make
things more complicated

External dependencies create bigger codebases and
islands of “magic”.

Web components are
nothing complicated, just
a few building blocks

Custom Elements API, Shadow DOM and template tags.
That's it. Use vanilla JS for the rest.

Web components in the browser

Custom Elements

```
customElements.define  
( 'tag-name', Class)
```

Lifecycle methods

Shadow DOM

```
this.attachShadow()
```

Encapsulation

Template tags

```
<template></template>
```

HTML reuse

Rest of the DOM API: `querySelector`, `addEventListener`, etc.
More APIs: `fetch`, `workers`, etc.

ES modules: `import * from 'anywhere'`, etc.

Vanilla JS: `string literals`, `Array.map`, etc.

Browser

Web components are freaking fun!

Also, they enable you to design code the way the web platform was designed to be used.

DEMO TIME!

Web components have been
around for a long time
and are picking up speed

Accelerated Mobile Pages, Polymer, Stencil and e.g.
Salesforce Lightning Web Components are all based on web
components.

Write vanilla
HTML/JS/CSS, use native
APIs and

have fun experimenting!

Thank you!

// questions welcome

`github.com/ericselin/web-components-sthlmjs`

`google.com/search?q=web+components`

`developer.mozilla.org/en-US/docs/Web/Web_Components`

`developers.google.com/web/fundamentals/web-components/
webcomponents.org`

`ericselin.dev`

`github.com/ericselin`

`gitlab.com/ericselin`

`glitch.com/@ericselin`

`twitter.com/ericselin`