

STENCIL

Universal Web Components Mike Hartington | @mhartington

GOALS

History lesson

Current events

Future is bright

A LONG TIME AGO...

Components on the Web

Known by many different names

Mostly built in

Custom ones...require JS

Early Players in this landscape include

jQuery

Everyone's first "framework"

Made web dev less hostile

jQuery Plugins: first "component" model

Hello World: Jquery



Open Sandbox

Console 0

Problems 0



jQuery was simple

Plugin API was 🤪

Backbone

Structure and compositability

jQuery? You bet!

Concept of a "base"

C

C[↗]

Open Sandbox

Console

0

Problems

0

^

Paved the way for a lot of ideas we see today

No 2-way data binding

More modules, not components

DOM updates, not batched

MV* FRAMEWORK WARS

Knockout, Ember, AngularJS

All pushing similar concepts

Custom components are getting closer

FAST FORWARD TO TODAY

MVC → COMPONENTS

Angular, (P)React, Vue

Again, all similar ideas

Custom components all the way down

ANGULAR



Open Sandbox

Console 1

Problems 0



REACT



Open Sandbox

Console 0

Problems 0

React DevTools 0



VUE



Open Sandbox

Console 1

Problems 0



But that's not all...

**EMBER, PREACT, CHOOJS, SKATEJS,
CANJS, DIOJS, HYPERHTML,
HYPERAPP, VIPERHTML SURPLUS,
DOJO, POLYMER, LIT-HTML, LIT-
ELEMENT, LIT-HTML-ELEMENT,
STIMULUS, AURELIA, KNOCKOUT,
BACKBONE, MARIONETTE, RIOT,
INFERNO, SVELT, MARKO, MITHRIL,
ETC**

oh my

*Frameworks keep reinventing the wheel of with custom UI,
but make APIs and implementation details that are not
compatible with one another*

HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.

YEAH!



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

What if...

Enter Web Components



What ever are web components

Template, Custom Elements, Shadow DOM

API that can work with all the frameworks

custom-elements-everywhere.com

Isn't this a Google/Chrome thing?

So is this like Polymer

Not exactly, Polymer === Polyfill

C

C[↗]

Open Sandbox

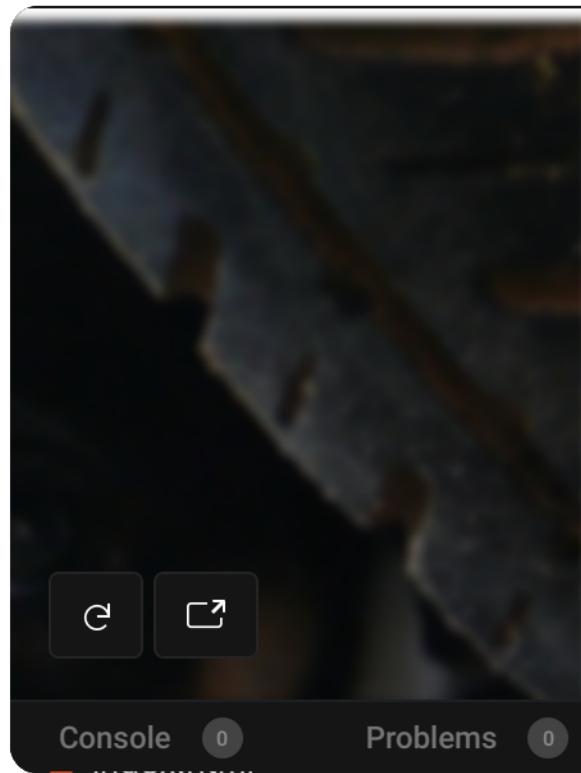
Console

0

Problems

0





C

ɔ

Console 0

Problems 0

Open Sandbox

^

pearlbea/better-img

Now this is nice...but feel lacking

Updates are not optimized

Properties are not "smart"

Overall: too low level



What is Stencil

A compiler for web components

Familiar APIs from across frameworks

Reactive Data binding

JSX/VDOM for speed

Stencil is not

A framework 😊

```
@Component({  
  tag: 'my-name',  
  styleUrls: ['my-name.scss']  
})  
export class MyName {  
  @Prop() name: string;  
  
  render() {  
    return <h1>Hello, my name is {this.name}</h1>;  
  }  
}
```

```
<my-name name="Mike"></my-name>
```

\$ DEMO TIME

API

@Component	Decorate a class a web component
@Prop()	Expose a class property as an element property
@State()	Internal data that can cause a rerender
@Event()	Decorator for custom events
@Listen()	Decorator for listening to custom events
@Element()	Get a raw DOM reference to the current component

So, why make Stencil

Interoperability

Performance

Stability

BUT WAIT, THERE'S MORE!

Frameworks on Web Components

Angular Elements: Wrap Angular components as WC



MAJOR PLAYERS AGREE

Common API is needed for frameworks

Web Components provide a primitive

Stencil Provides a slightly better abstraction

Still Web Components

Stencil in the wild today

Ionic

Porsche

Amazon

Apple

Web Components are here to stay

THANKS

Mike Hartington | @mhartington

`</html>`