{codemotion}

We code the future. Together

Devs Lunch Box

stencil

A quick introduction

Horacio Gonzalez
@LostInBrittany

{CODEMOTION}

stencil

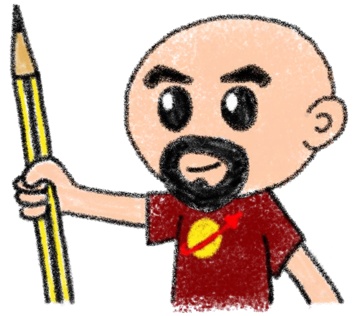
Stencil 101

Horacio Gonzalez
@LostInBrittany

OVHcloud

Who are we?

Introducing myself and introducing ~~OVH~~ OVHcloud



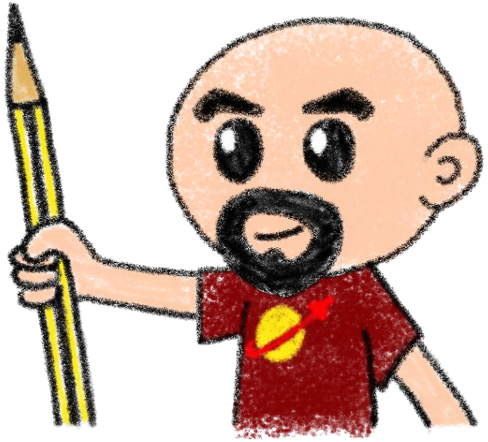
Horacio Gonzalez

@LostInBrittany

Spaniard lost in Brittany, developer, dreamer and all-around geek

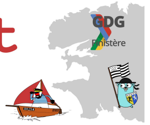
OVHcloud

DevRel Leader



DevFest du Bout du Monde

Finist Devs



Google Developers Experts 2019

Web Technologies GDE

Flutter

OVHcloud: A Global Leader

- **Web Cloud & Telcom**
- **Private Cloud**
- **Public Cloud**
- **Storage**
- **Network & Security**

30 Data Centers
in 12 locations

34 Points of Presence
on a 20 TBPS Bandwidth Network

2200 Employees
worldwide

115K Private Cloud
VMS running

300K Public Cloud
instances running

380K Physical Servers
running in our data centers

1 Million+ Servers
produced since 1999

1.5 Million Customers
across 132 countries

3.8 Million Websites
hosting

1.5 Billion Euros Invested
since 2016

P.U.E. 1.09
Energy efficiency indicator

20 Years in Business
Disrupting since 1999

The 3 minutes context

What the heck are web component?

Web Components

Web standard W3C

Web Components

Available in all modern browsers:

Firefox, Safari, Chrome

Web Components

Create your own HTML tags

Encapsulating look and behavior

Web Components

Fully interoperable

With other web components, with any framework

Web Components

CUSTOM ELEMENTS

SHADOW DOM

TEMPLATES

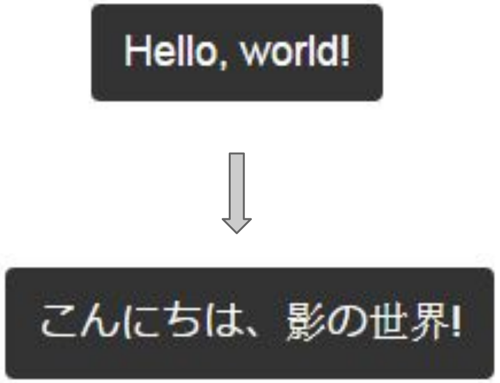
Custom Element

</> To define your own HTML tag

```
<body>
  ...
  <script>
    window.customElements.define('my-element',
            class extends HTMLElement {...});
  </script>
  <my-element></my-element>
</body>
```

Shadow DOM

To encapsulate subtree and style in an element



```
<button>Hello, world!</button>
<script>
var host = document.querySelector('button');
const shadowRoot = host.attachShadow({mode:'open'});
shadowRoot.textContent = 'こんにちは、影の世界!';
</script>
```

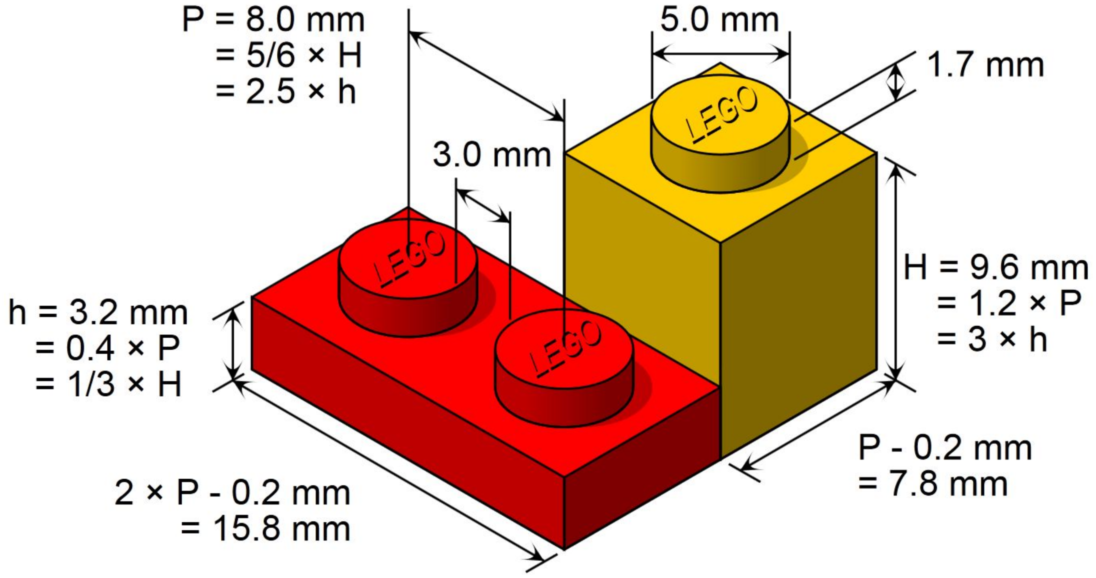
Template

To have clonable document template

```
<template id="mytemplate">
  <img src="" alt="great image">
  <div class="comment"></div>
</template>

var t = document.querySelector('#mytemplate');
// Populate the src at runtime.
t.content.querySelector('img').src = 'logo.png';
var clone = document.importNode(t.content, true);
document.body.appendChild(clone);
```

But in fact, it’s just an element...

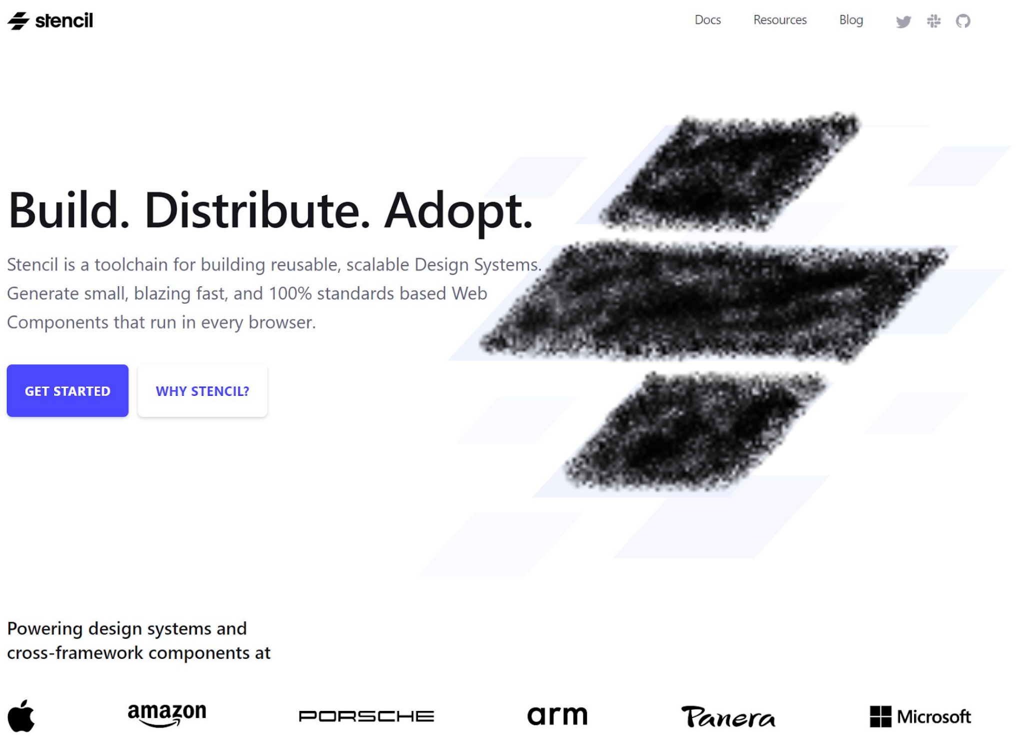
- Attributes
- Properties
- Methods
- Events



Stencil

Powering Ionic 4+

Not another library



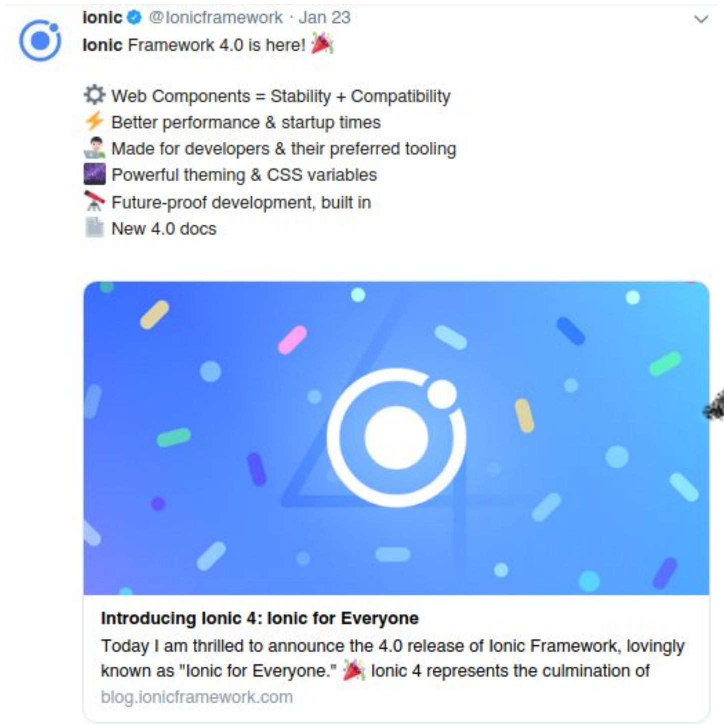
Build. Distribute. Adopt.

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.

Powering design systems and cross-framework components at

A Web Component toolchain

A mature technology

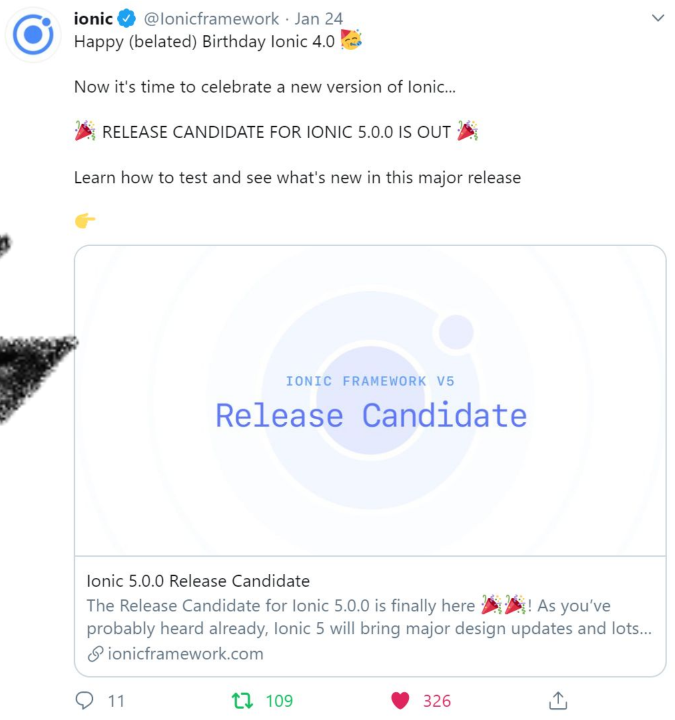


ionic @Ionicframework · Jan 23

Ionic Framework 4.0 is here! 🎉

⚙️ Web Components = Stability + Compatibility
⚡ Better performance & startup times
👨‍💻 Made for developers & their preferred tooling
🌌 Powerful theming & CSS variables
🔭 Future-proof development, built in
📄 New 4.0 docs

Introducing Ionic 4: Ionic for Everyone
Today I am thrilled to announce the 4.0 release of Ionic Framework, lovingly known as "Ionic for Everyone." 🎉 Ionic 4 represents the culmination of
blog.ionicframework.com



ionic @Ionicframework · Jan 24

Happy (belated) Birthday Ionic 4.0 🥳

Now it's time to celebrate a new version of Ionic...

🎉 RELEASE CANDIDATE FOR IONIC 5.0.0 IS OUT 🎉

Learn how to test and see what's new in this major release

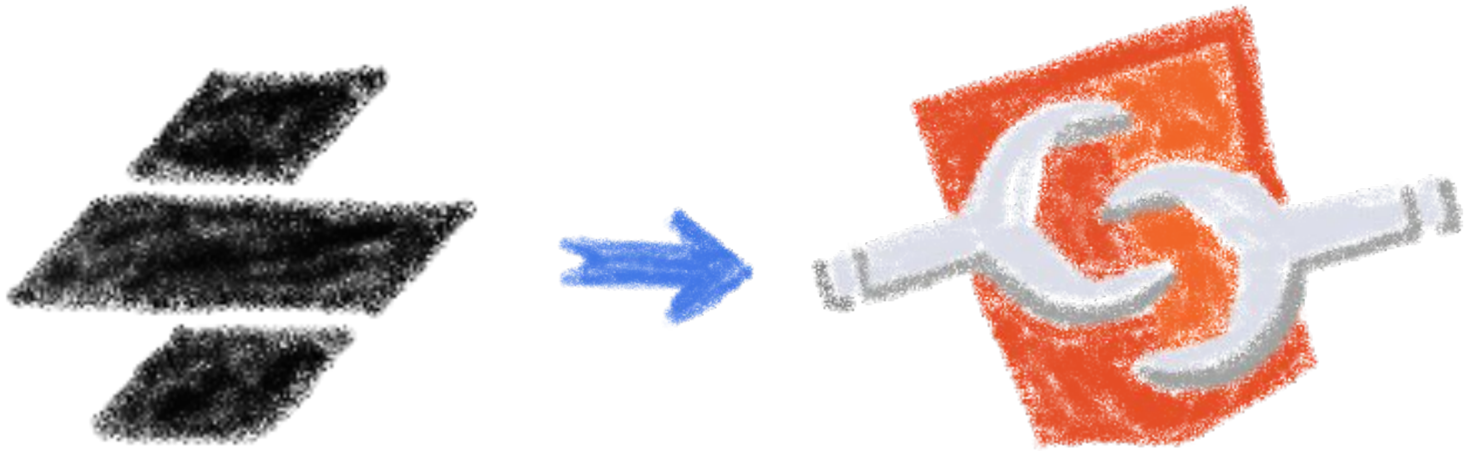
👉

Ionic 5.0.0 Release Candidate
The Release Candidate for Ionic 5.0.0 is finally here 🎉🎉! As you've probably heard already, Ionic 5 will bring major design updates and lots...
ionicframework.com

11 109 326

Ionic 4 released on year ago, powered by Stencil!

A build time tool

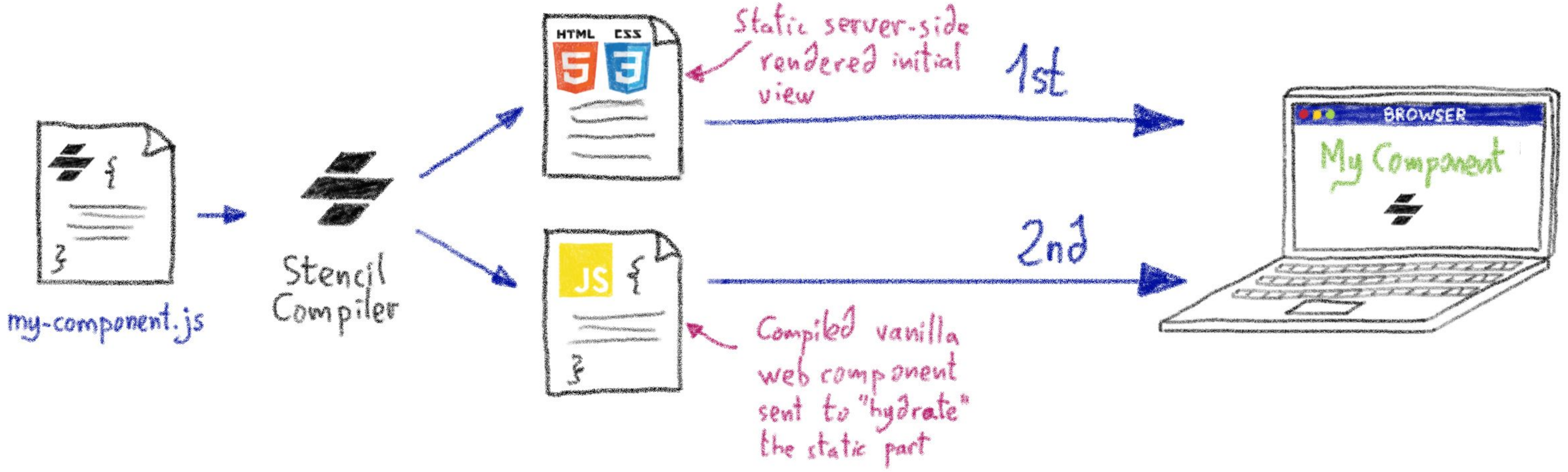


To generate standard web components

Fully featured

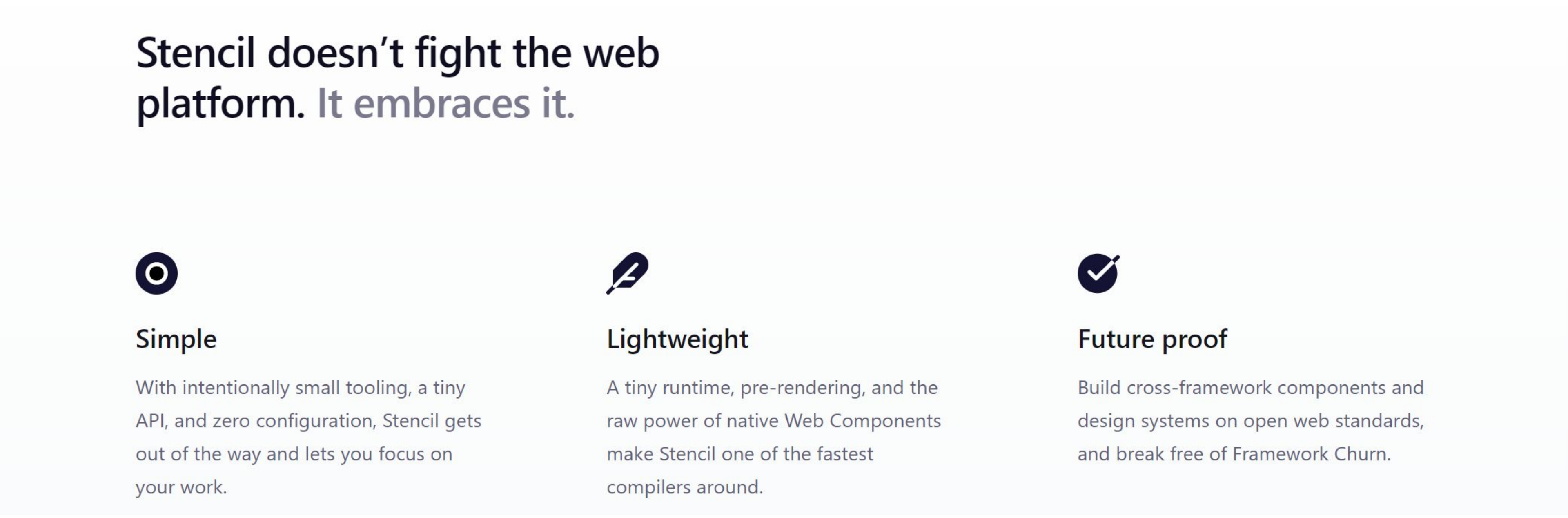
- Web Component-based
- Asynchronous rendering pipeline
- TypeScript support
- Reactive Data Binding
- Component pre-rendering
- Simple component lazy-loading
- JSX support
- Dependency-free components

And the cherry on the cake


my-component.js
Stencil Compiler
HTML
CSS
Static server-side rendered initial view
1st
JS
2nd
Compiled vanilla web component sent to "hydrate" the static part
BROWSER
My Component

Server-Side Rendering

Stencil leverages the web platform



Stencil doesn't fight the web platform. It embraces it.

Simple

With intentionally small tooling, a tiny API, and zero configuration, Stencil gets out of the way and lets you focus on your work.

Lightweight

A tiny runtime, pre-rendering, and the raw power of native Web Components make Stencil one of the fastest compilers around.

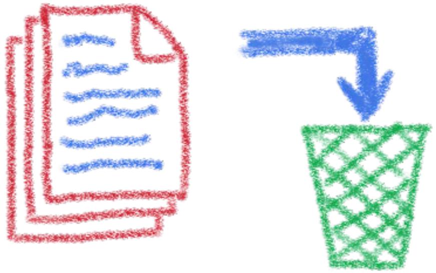
Future proof

Build cross-framework components and design systems on open web standards, and break free of Framework Churn.

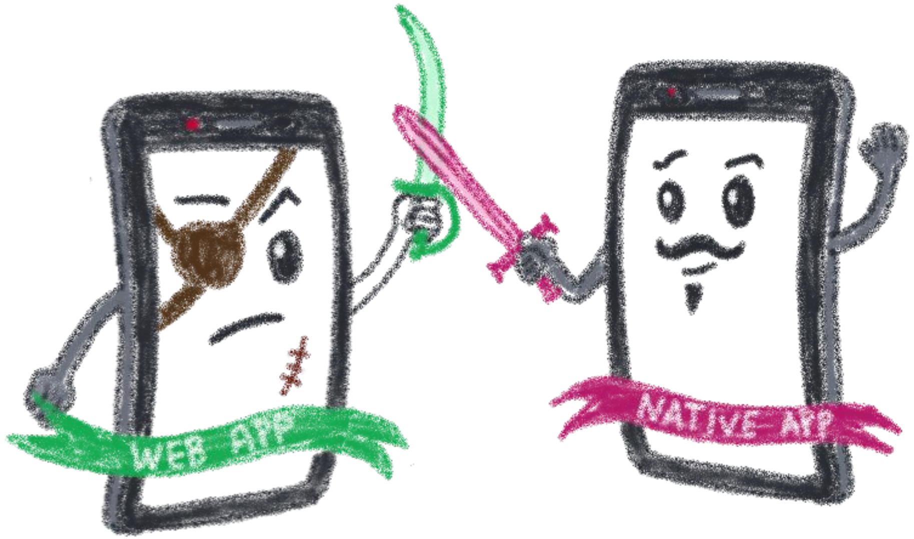
Working with the web, not against it

The Stencil story

A company tired of putting good code in the bin

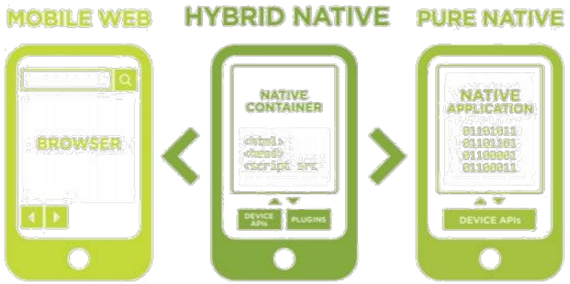


Once upon a time there was a fight



Between native apps and web app on mobile

A quest to the perfect solution

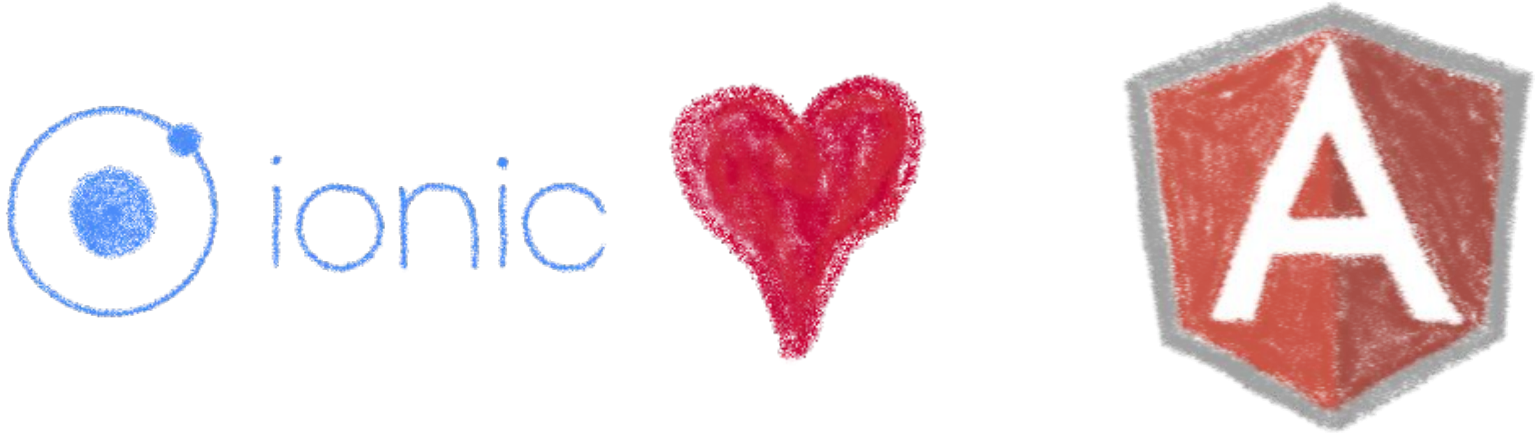


Hybrid apps, leveraging on web technologies

A company wanted to do it well

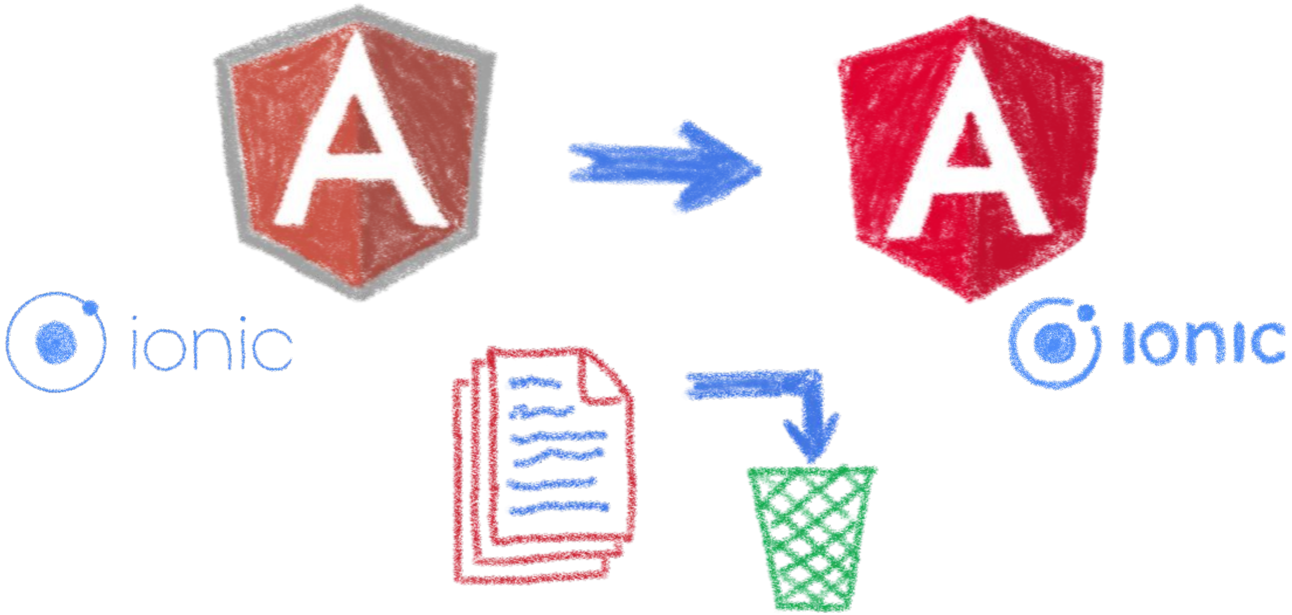
The perfect technology for mobile web and hybrid apps

The time is 2013



So what technology would you use?

Really soon after launch...



Hey folks, we are killing AngularJS!

What did Ionic people do?

Let's put everything in the trash bin and begin anew

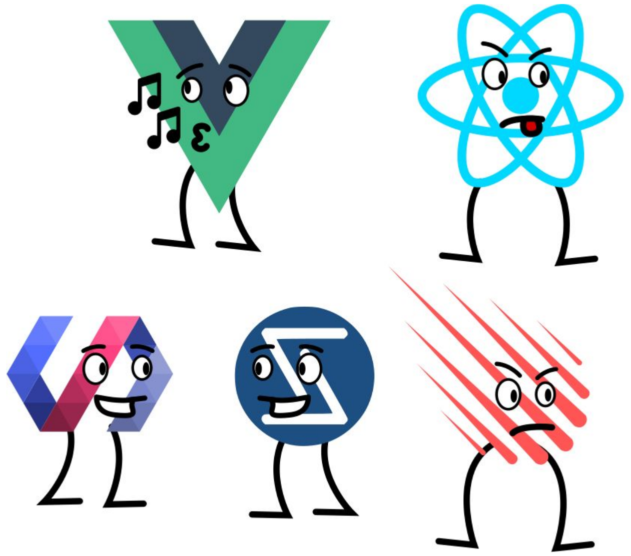
But times have changed...

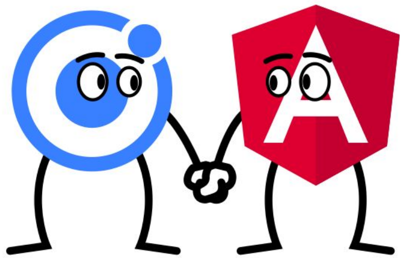
In 2013 Angular JS was the prom queen

Times have changed...

In 2017 Angular is only one more in the clique

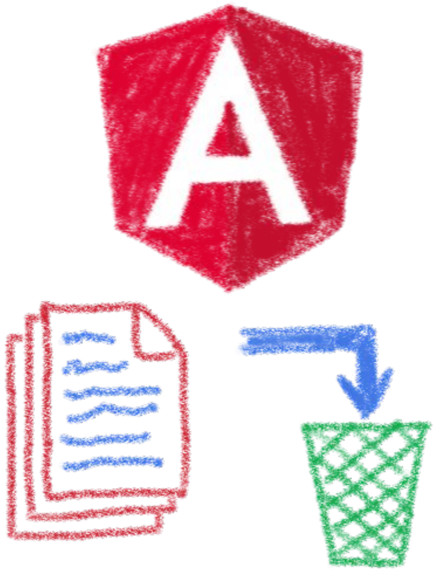
Angular limits adoption of Ionic





Devs and companies are
very vocal about JS Frameworks

What did Ionic people do?



Let's put everything in the trash bin and begin anew...

But on which framework?

What about web components?

A nice solution for Ionic problems:

Any framework, even no framework at all!

But what Web Component library?

hybrids

snuggsi ツ

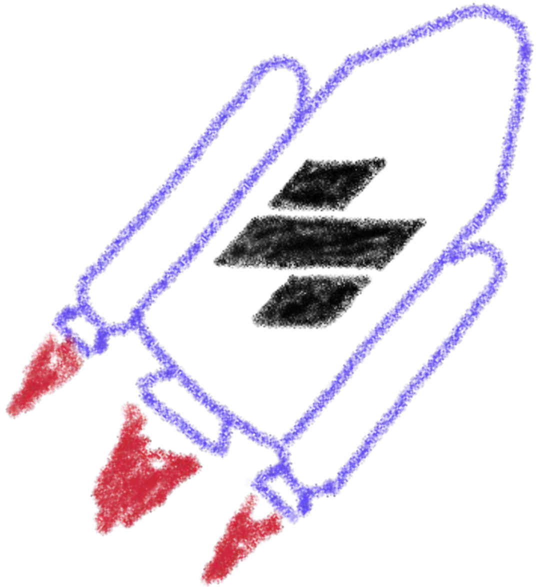
smart HTML ELEMENTS

SkateJS

slim.js

There were so many of them!

Let's do something different



A fully featured web component toolchain

With all the bells and whistles!

Ionic rewrote all their code again

Ionic 4 is fully based on Ionic

Now Ionic works on any framework

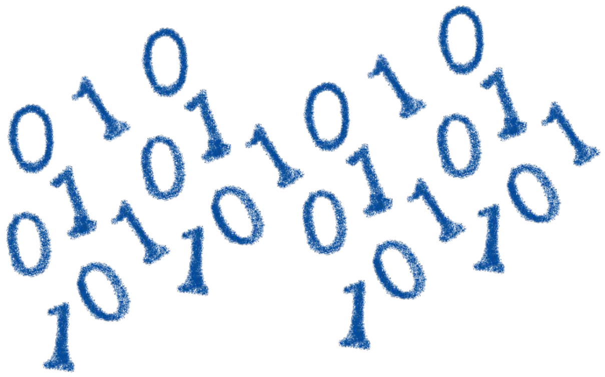
Or without framework at all

And we have Stencil

To use it in any of our projects

Hey dude, enough stories!

We are here to see some code!



Hands on Stencil

Simply use npm init

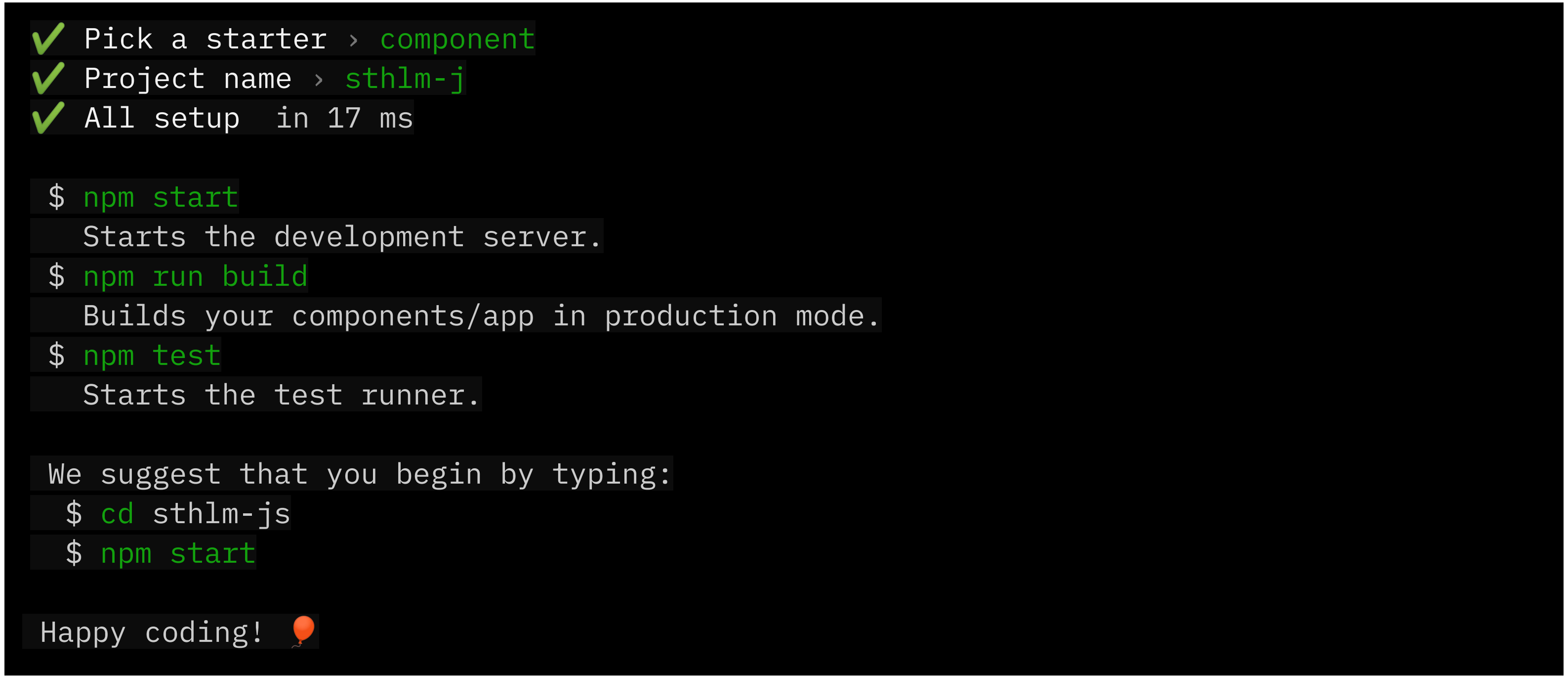
```
npm init stencil
```

Choose the type of project to start

```
? Pick a starter › - Use arrow-keys. Return to submit.

❯  ionic-pwa     Everything you need to build fast, production ready PWAs
   app           Minimal starter for building a Stencil app or website
   component     Collection of web components that can be used anywhere
Updating Stencil
```

Hands on Stencil

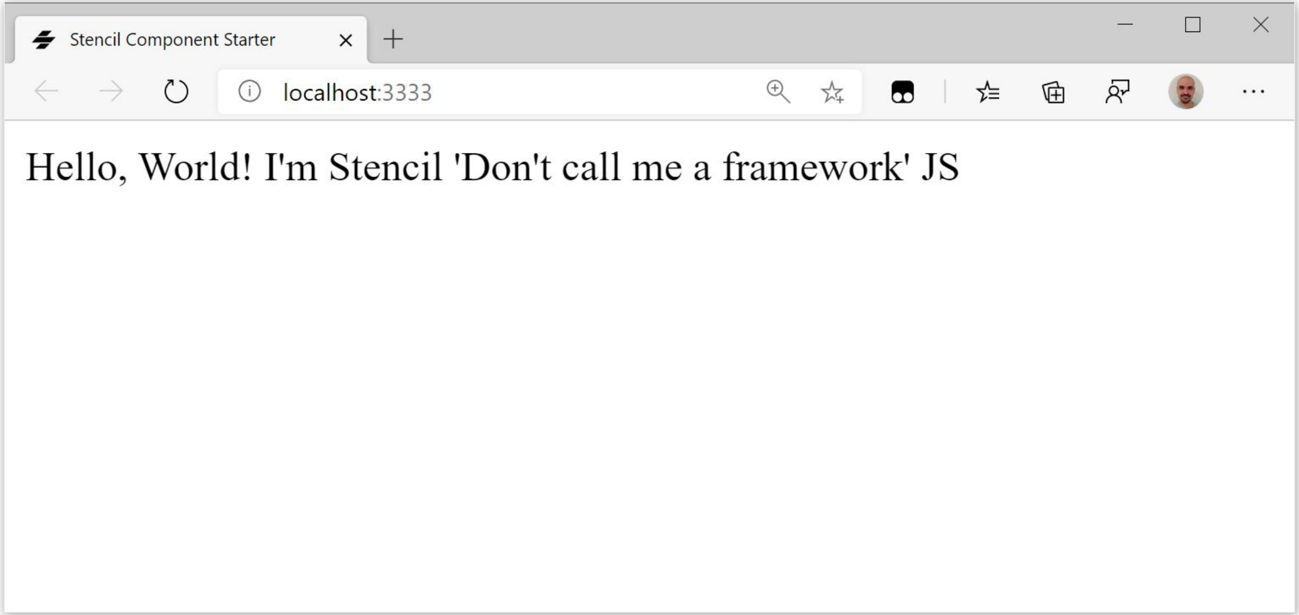
And the project is initialized in some seconds!



```
✓ Pick a starter › component
✓ Project name › sthlm-j
✓ All setup  in 17 ms

 $ npm start
   Starts the development server.
 $ npm run build
   Builds your components/app in production mode.
 $ npm test
   Starts the test runner.

 We suggest that you begin by typing:
  $ cd sthlm-js
  $ npm start

Happy coding! 🎈
```

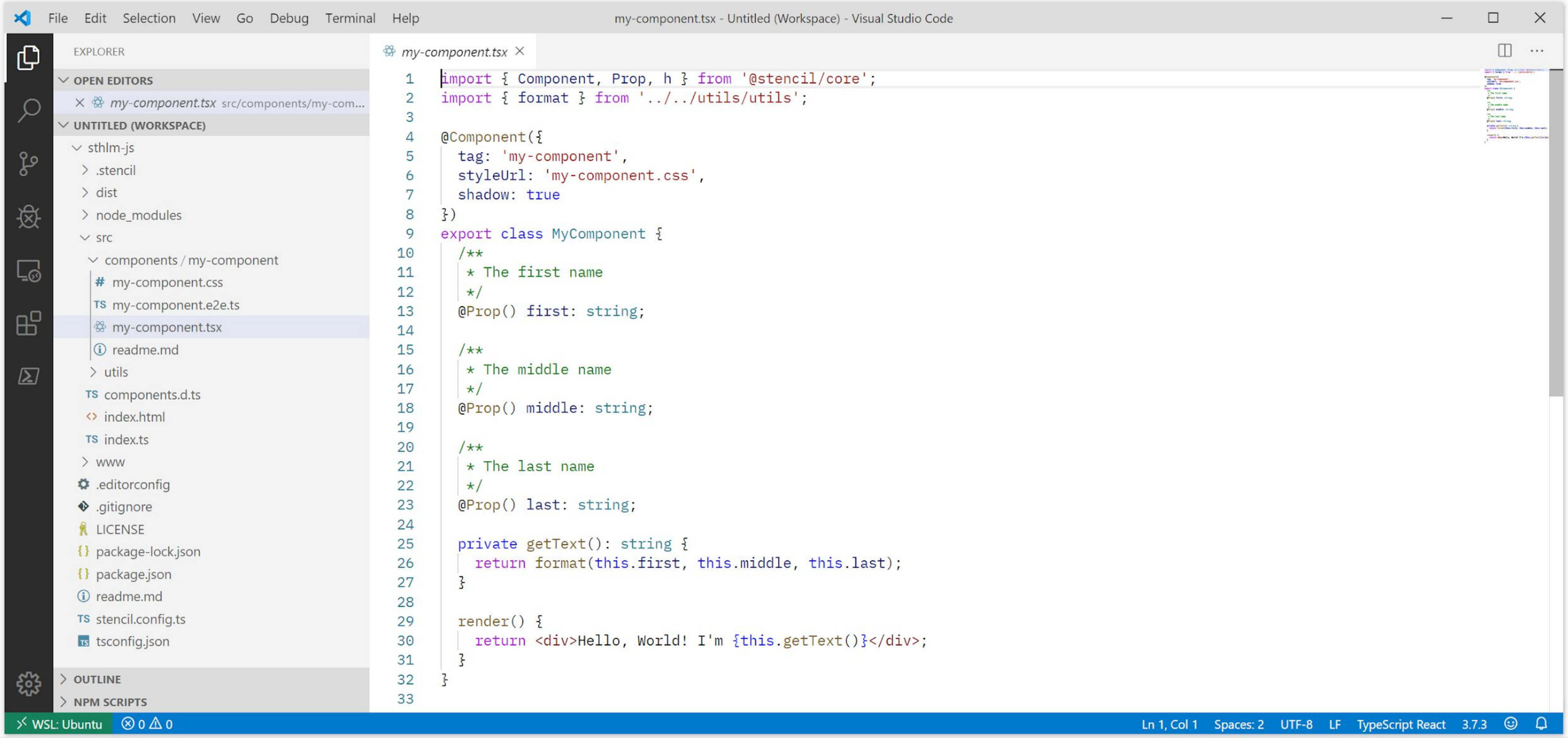
Starting the development server

```
npm start
```



Hello, World! I'm Stencil 'Don't call me a framework' JS

Let's look at the code



```tsx
import { Component, Prop, h } from '@stencil/core';
import { format } from '../../utils/utils';

@Component({
  tag: 'my-component',
  styleUrl: 'my-component.css',
  shadow: true
})
export class MyComponent {
  /**
   * The first name
   */
  @Prop() first: string;

  /**
   * The middle name
   */
  @Prop() middle: string;

  /**
   * The last name
   */
  @Prop() last: string;

  private getText(): string {
    return format(this.first, this.middle, this.last);
  }

  render() {
    return <div>Hello, World! I'm {this.getText()}</div>;
  }
}

```

Some concepts

```
import { Component, Prop, h } from '@stencil/core';
import { format } from '../../utils/utils';

@Component({
  tag: 'my-component',
  styleUrl: 'my-component.css',
  shadow: true
})
export class MyComponent {

  @Prop() first: string;
```

Decorators

Some concepts

```
@Prop() first: string;

@Prop() middle: string;

@Prop() last: string;

@State() nickname: string;
```

Properties and States

Some concepts

```
render() {
  return <div>Hello, World! I'm {this.getText()}</div>;
}
```

Asynchronous rendering using JSX

Some concepts

```
@Prop() value: number;

@Watch(value)
valueChanged(newValue: boolean, oldValue: boolean) {
  console.log(`The new value is ${newValue}, it was ${oldValue} before`);
}
```

Watch

Some concepts

```
@Event() actionCompleted: EventEmitter;

someAction(message: String) {
  this.actionCompleted.emit(message);
}
```

Emitting events

```
@Listen('actionCompleted')
actionCompletedHandler(event: CustomEvent) {
  console.log('Received the custom actionCompleted event: ', event.detail);
}
```

Listening to events

Some concepts

```
@Method()
async sayHello() {
  this.hello = true;
}

render() {
  return (
    <Host>
      <h2>{ this.hello ? `Hello sthlm.js` : ''}</h2>
    </Host>
  );
}
```

Asynchronous public methods

Some concepts

```
@Component({
  tag: 'my-component',
  styleUrl: 'my-component.css',
  shadow: true
})
export class MyComponent {
```

Optional Shadow DOM

Stencil for design systems

Because web components really shine for that

What the heck is a design system?

Components

Patterns

No more time spent rewriting once again the same base UI elements

Visual language

Design artifacts & Code implementation

Why Stencil is so good for design systems?

Web Components work everywhere!

One more thing...*

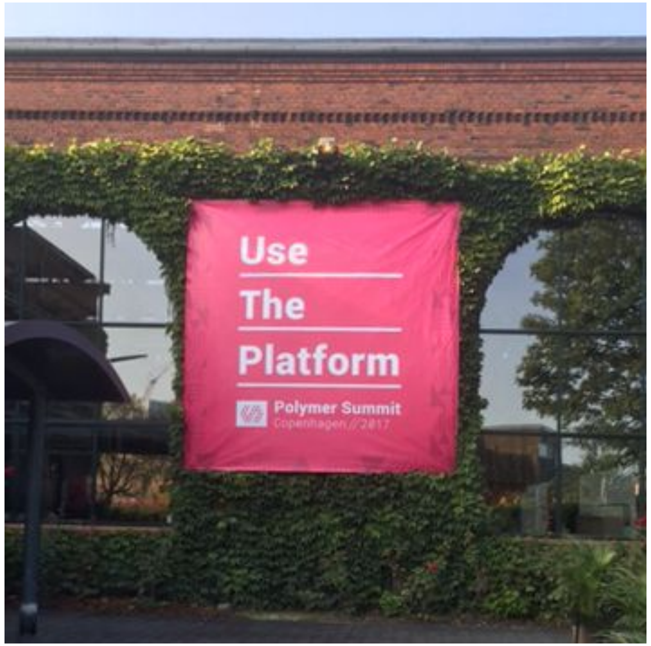
Let's copy from the master

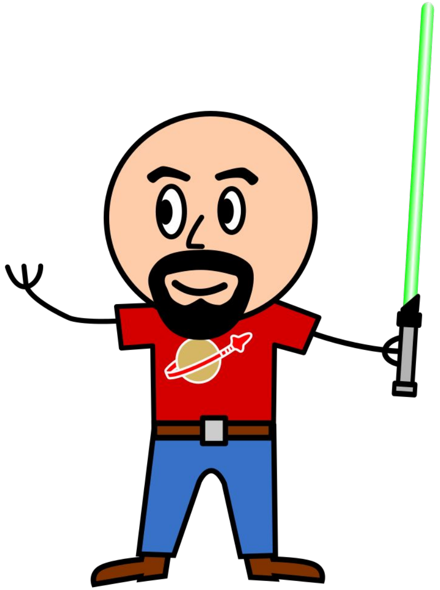
Stencil is not so important

WHEN THE SAGE POINTS AT THE MOON, THE FOOL LOOKS AT THE FINGER.
- BUDDHIST PROVERB
@AmericnDreaming
AmericanDreaming.deviantart.com

WebComponents ARE

Use the Platform, Luke...





WebComponents ARE native

Do you love your framework?

Oh yeah, we all do

Would you marry your framework?

Like until death...

How much does cost the divorce?

Do you remember when you dropped AngularJS for Angular?

Why recode everything again?

Reuse the bricks in your new framework

Lots of web components libraries

slim.js

LitElement

hybrids

snuggsi ツ

SkateJS

smart HTML ELEMENTS

stencil

For different need and sensibilities

And some good news

Angular Elements

Vue Web Component Wrapper

Frameworks begin to understand it

So for your next app

Choose a framework, no problem...

But please, help your future self

Use Web Components!

Conclusion

That's all, folks!