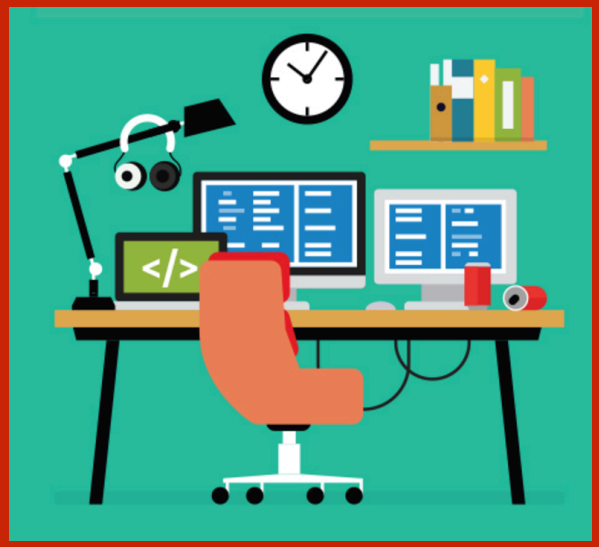
<ng-India/>

Demystifying Web Components

Perfect Weapon for the Convergence of Web

ng-India



Rahat Khanna

@mappmechanic

Currently @ Apple

Ex- Flipkart, Genpact

#ngIndia

www.ng-ind.com

Feb 24, 2018

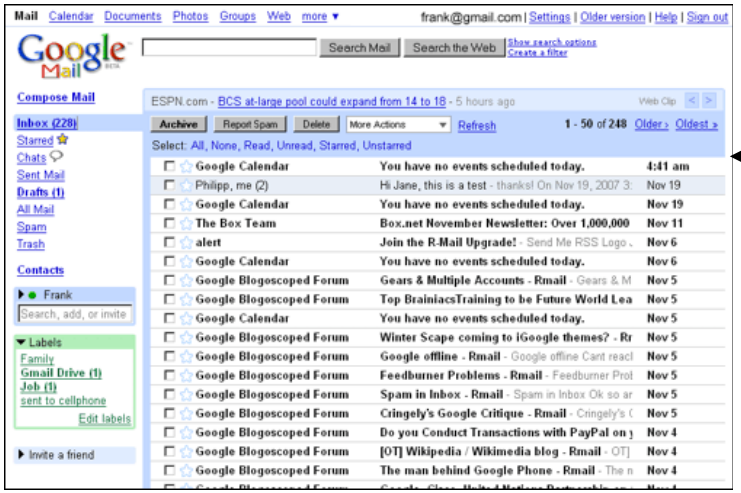
Agenda

- Problems in the current Front End World
- Getting Started with Web Components
- 4 Pillars of Web Components

- Using a natively developed Web Component with an Angular App

<ng-India/>

Evolution of Front End Dev

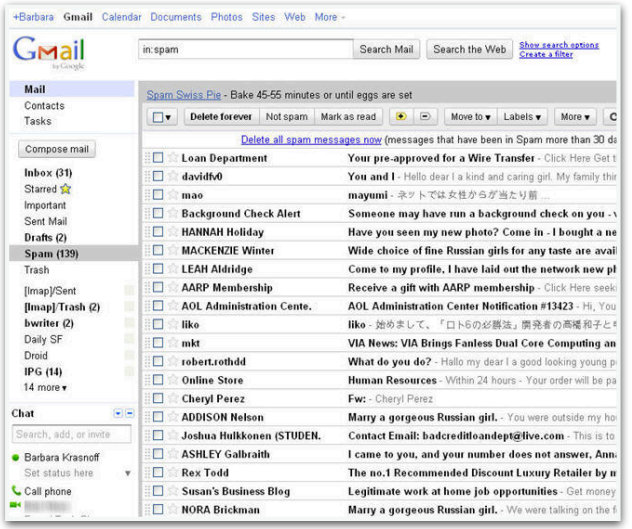


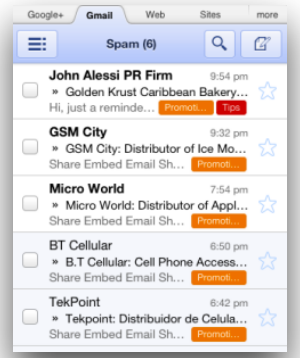
Static HTML + CSS

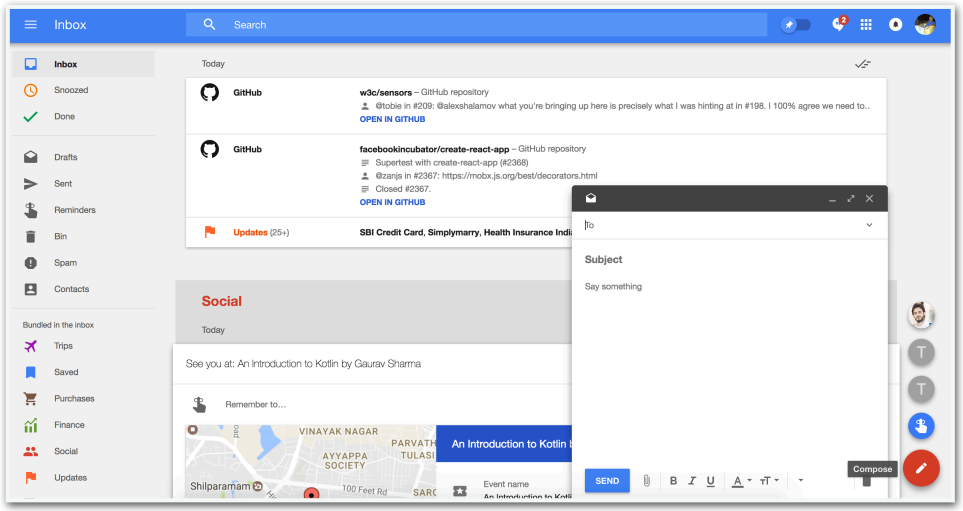
Served from Server

Ajax, DHTML, CSS

Mobile Web, Responsive







SPA, UI/UX, CSS3 Animations

PWAs, Server Rendering,

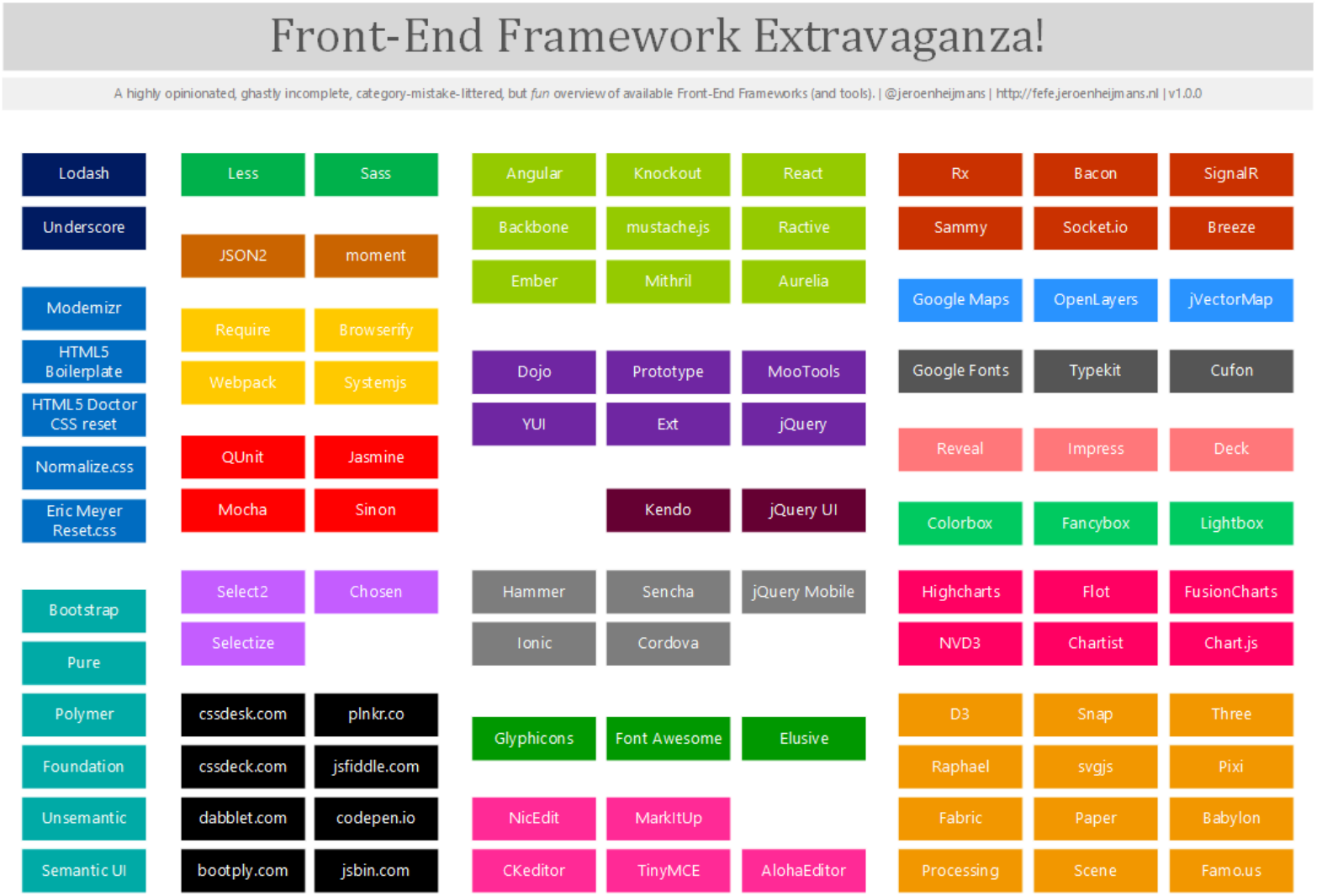
Bundling, CDN

<ng-India/>

Problems

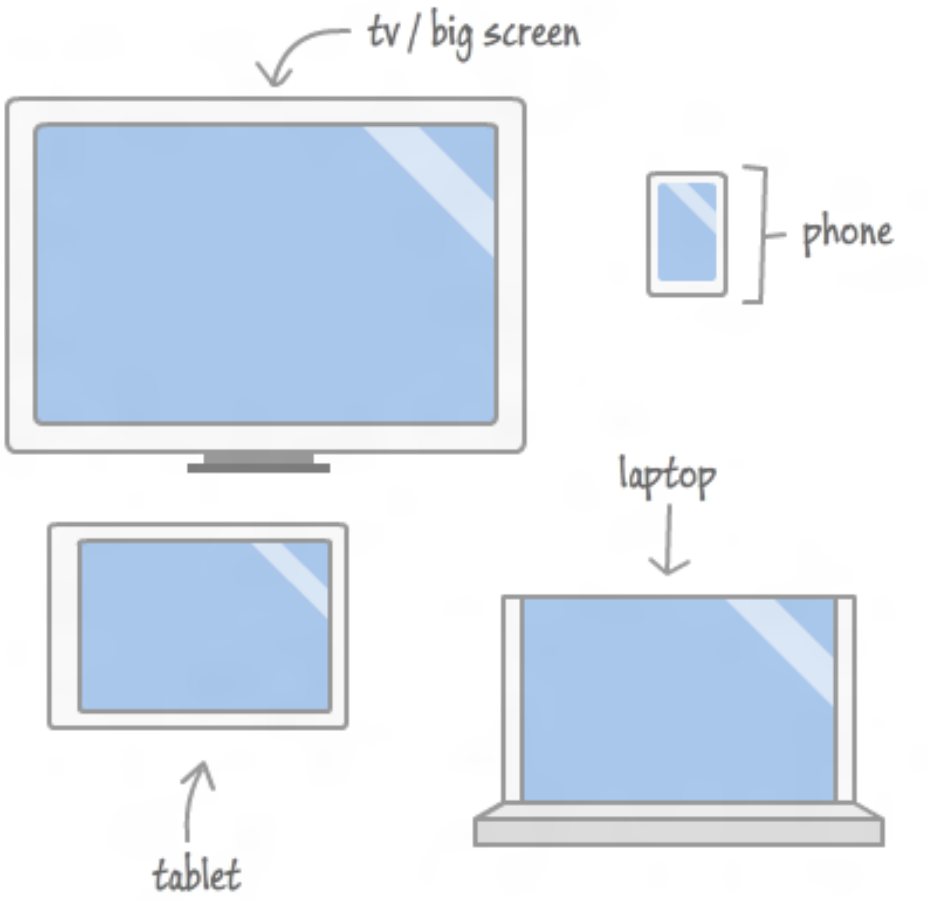
Problems

1. Learn a new framework every 6 months



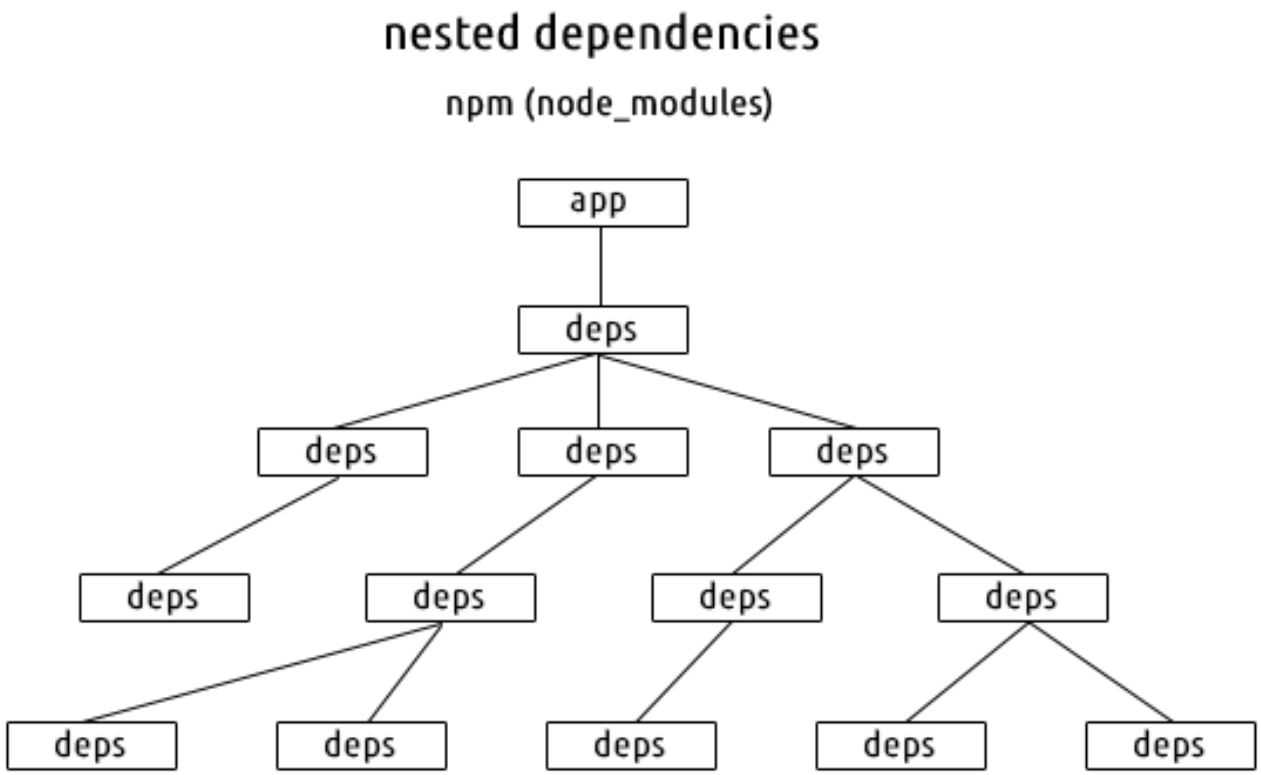
Problems

2. Device sizes and Browser quirks



Problems

3. Dependency management - node js dependency hell

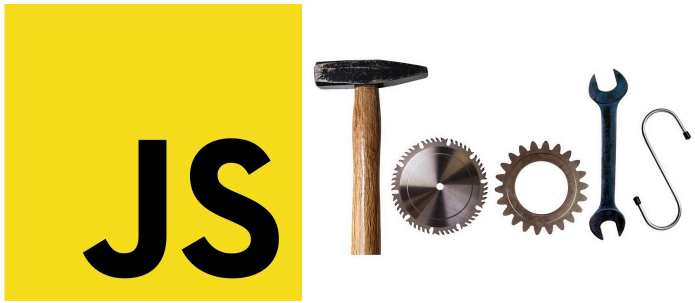


Problems

4. Build Management & deployment to CDNs

```
Version: webpack 1.12.9
Time: 31ms
     Asset       Size  Chunks             Chunk Names
index.html  165 bytes          [emitted]

ERROR in ./app/app.js
Module parse failed: c:\MMDATA\Gonzalo-Dev\gonzalobarba.com\app\app.js Line 1: Unexpected token
You may need an appropriate loader to handle this file type.
| import React, {Component} from 'react';
| import {render} from 'react-dom';
|
webpack: bundle is now VALID.
```



or

Problems

5. Referencing of HTML5, CSS & JS separately for reusable components

```
<head>
    <title>Angularjs Component</title>
    <meta charset="UTF-8">
    <script type="text/javascript" src="lib/jquery/jquery-2.1.1.min.js"></script>
    <script type="text/javascript" src="lib/bootstrap/js/bootstrap.min.js"></script>
    <script type="text/javascript" src="lib/angularjs/angular.min.js"></script>
    <script type="text/javascript" src="lib/angularjs/angular-resource.min.js"></script>
    <script type="text/javascript" src="lib/angularjs/angular-animate.min.js"></script>
    <script type="text/javascript" src="lib/angularui/ui-bootstrap-tpls-0.11.0.min.js"></script>
    <script type="text/javascript" src="lib/angularui/angular-ui-router.min.js"></script>
    <script type="text/javascript" src="lib/angularui/ng-grid/ng-grid-2.0.11.min.js"></script>
    <script type="text/javascript" src="lib/underscore/underscore-min.js"></script>
    <script type="text/javascript" src="lib/utils/spin.min.js"></script>
    <script type="text/javascript" src="lib/utils/jquery.bpopup.min.js"></script>

    <script type="text/javascript" src="js/app.js"></script>
    <script type="text/javascript" src="js/controllers.js"></script>
    <script type="text/javascript" src="js/directives.js"></script>
    <script type="text/javascript" src="js/filters.js"></script>
    <script type="text/javascript" src="js/services.js"></script>
    <script type="text/javascript" src="js/utils.js"></script>
    <script type="text/javascript" src="config/config.js"></script>
    <script type="text/javascript" src="test/restful/restful.js"></script>

    <!-- tabs controller -->
    <script type="text/javascript" src="js/navs/navs.js"></script>
    <script type="text/javascript" src="js/navs/nav1.js"></script>
    <script type="text/javascript" src="js/navs/nav2.js"></script>
    <script type="text/javascript" src="js/navs/nav3.js"></script>

    <link rel="stylesheet" href="lib/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="lib/angularui/ng-grid/ng-grid.min.css">
    <link rel="stylesheet" href="css/app.css">
</head>
```

Problems are not **STOP** signs,
they are **guidelines**

<ng-India/>

You already know Web Components ? Don't you ?

<ng-India/>

Web Components as Weapon for Web Convergence

<ng-India/>

History of Web Components

2013 - first mentioning of WC by Google - v0

2014 - added official support v0 in Chrome & Opera

2015 - several meetings to discuss what goes to v1 and what to further versions

2016 - added official support of custom elements v1 in Chrome and Opera

2017 - Shadow DOM support added in Safari (desktop & mobile) and is coming to Firefox soon.

Architecture

JS Code for Component Behaviour

ES6 Classes (mostly)

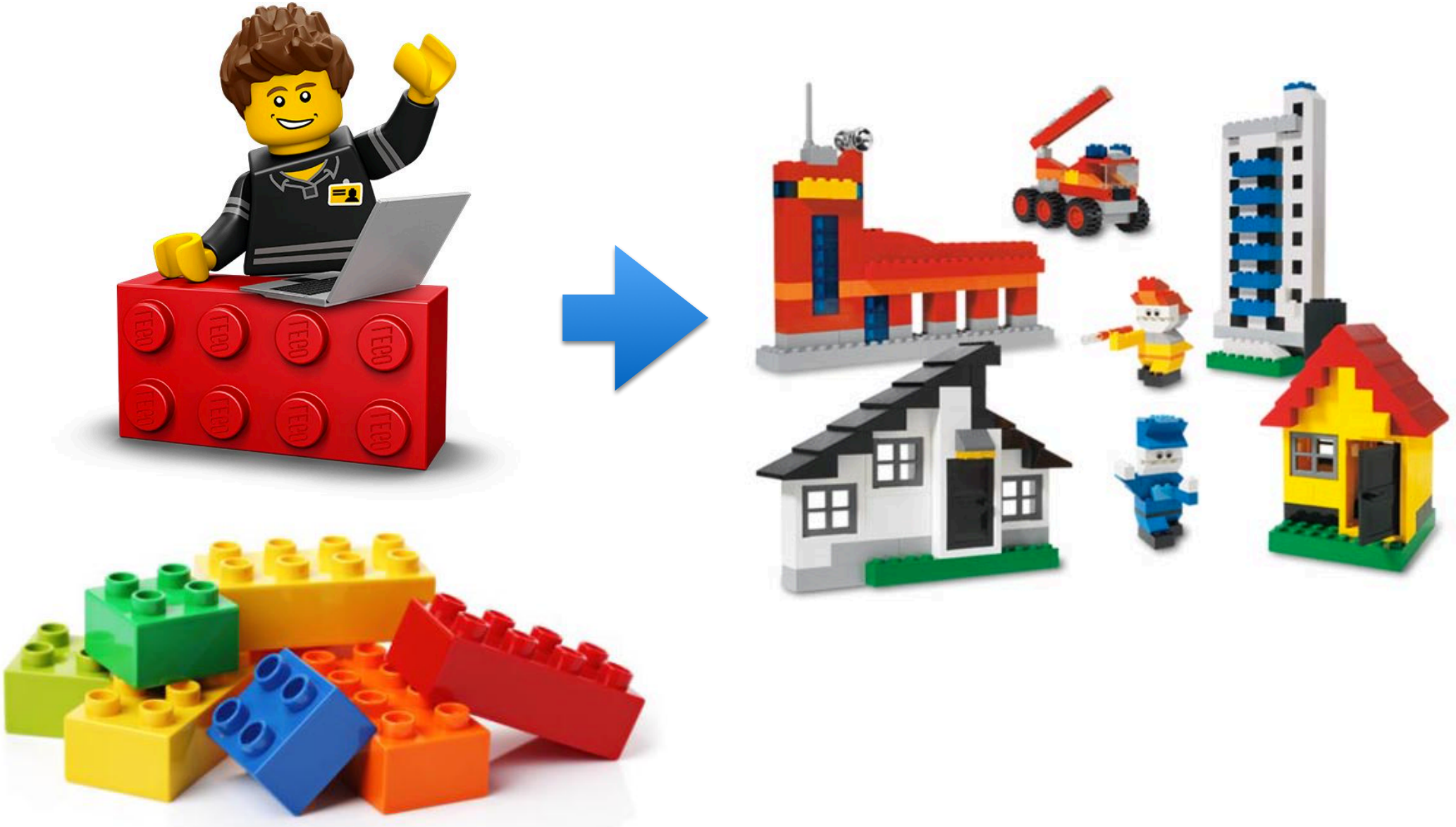
Web Components Code

HTML Templates + CSS Styles

Polyfills like PolymerJS, X-Tag which are temporary

Browser APIs - like Custom Elements v1, Shadow DOM, HTML Imports

Why & Where should we use Web Components ?



Custom Elements

Create new HTML tags or **Extend** existing ones
Foundation of web components

```
<my-app>
    <my-header>
        <h1>App Title</h1>
    </my-header>
    <my-content>
        <form>
            Enter name<input type="text">
            <my-button>Submit</my-button>
        </form>
    </my-content>
</my-app>
```

<ng-India/>

Creating a Simple Custom Element

```
// ES2015 Class for the new Custom Element
class MyButton extends HTMLElement {
    constructor() {
        console.log('my button has been initialised.');
    }

    // Getter & Setter for Properties
    get color() {
        return this._color;
    }

    set color(value) {
        // Reflect the value of the open property as an HTML attribute.
        if (val) { this.setAttribute('color', value); } else { this.removeAttribute('color'); }
        this.updateColor();
    }

    updateColor() {
    }
}

customElements.define('my-button', MyButton);
```

Lifecycle Methods - Callbacks

constructor()
Called when the element is created or upgraded

connectedCallback()
Called when the element is inserted into a document, including into a shadow tree

disconnectedCallback()
Called when the element is removed from a document

attributeChangedCallback(attributeName, oldValue, newValue, namespace)
Called when an attribute is changed, appended, removed, or replaced on the element. Only called for observed attributes.

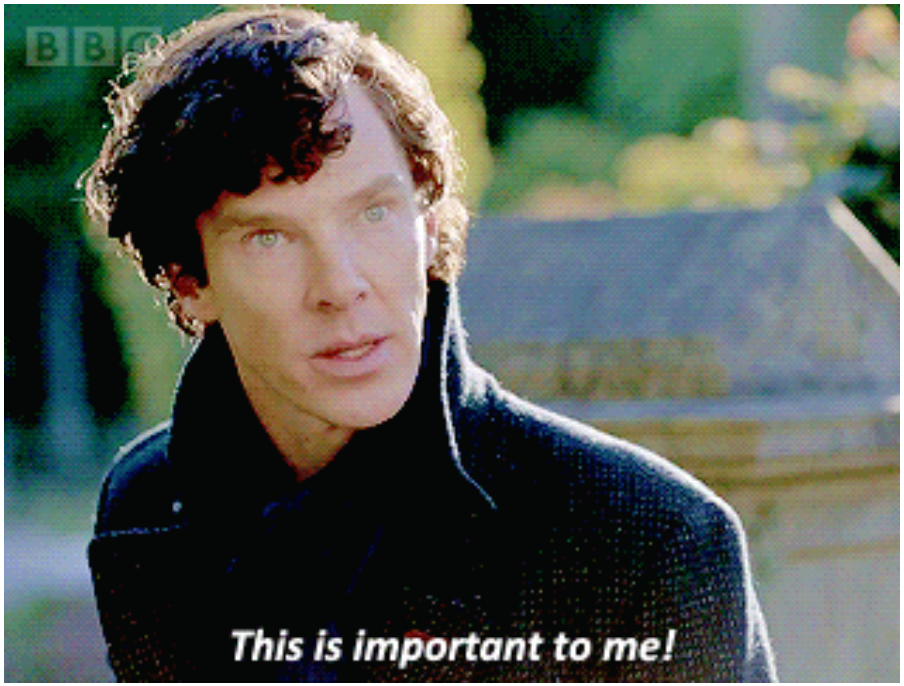
adoptedCallback(oldDocument, newDocument)
Called when the element is adopted into a new document

hands on

Shadow DOM

Shadow DOM is a new specification which will help developers to keep **some HTML & CSS code separate from the rest of the page.**

color: #fff **!important**;



Benefits of Shadow DOM

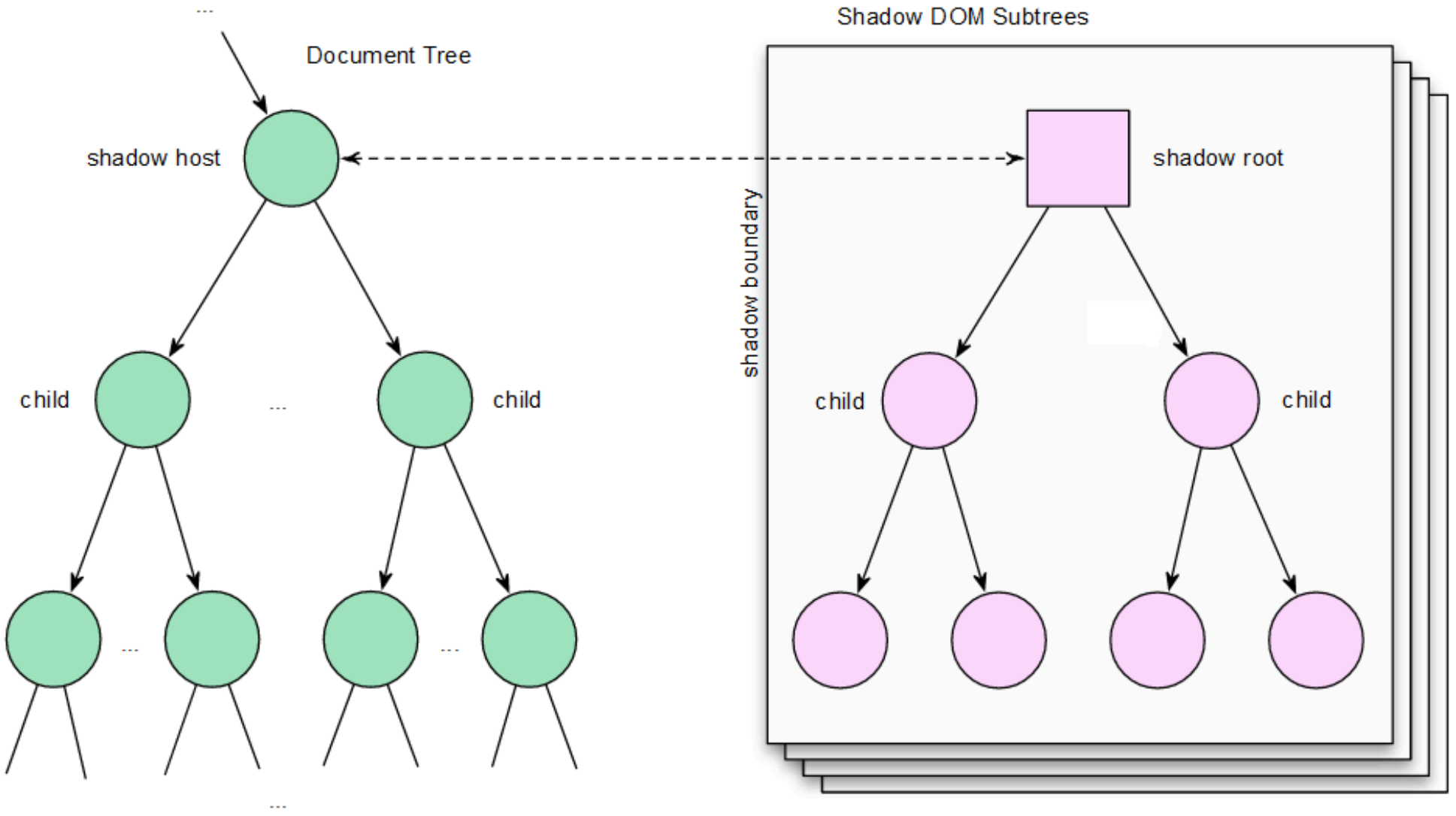
- **Isolated DOM** - self contained component DOM
- **Scoped CSS** - css defined inside shadow DOM
- **Composition** - declarative markup with hidden DOM
- **Simplifies CSS** - simple css selectors without conflicts
- **Productivity** - build using reusable components

Shadow DOM (contd)

- Shadow DOM is attached to an existing element
  - Either a native element like or <p>
  - Or a Custom Element like <my-element>

- Style Shadow DOM like normal CSS and few extra selectors
  - Pseudo-classes- :host, :host-context()
  - Pseudo-elements- ::slotted()
  - Combinator- >>>*

Shadow DOM Diagram



HTML Template Element

The **HTML `<template>` element** is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.

HTML Template Element

Example

```
<table id="producttable">
  <thead>
    <tr>
      <td>UPC_Code</td>
      <td>Product_Name</td>
    </tr>
  </thead>
  <tbody>
    <!-- existing data could optionally be included here -->
  </tbody>
</table>

<template id="productrow">
  <tr>
    <td class="record"></td>
    <td></td>
  </tr>
</template>
```

Slot Element - Light DOM

```
<slot> </slot>
```

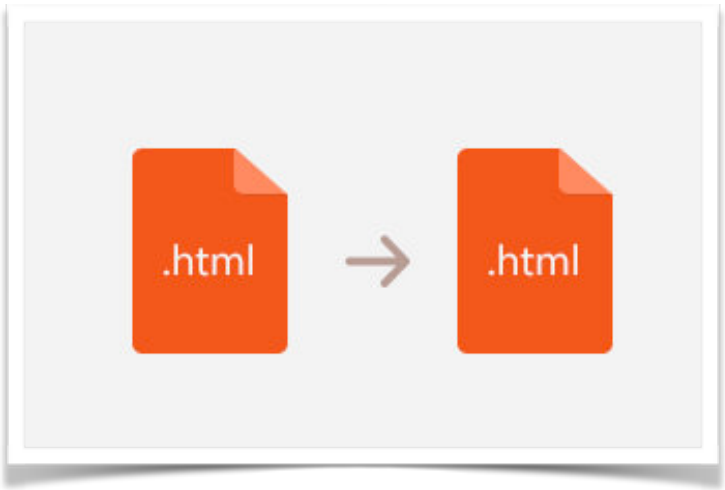
```
<slot name='icon'> </slot>
```

`<slot>` is a placeholder where child nodes will render

HTML Imports (not recommended)

HTML Imports are a way to include and reuse HTML documents in other HTML documents.

```
<link href="extern.html" rel="import" />
```

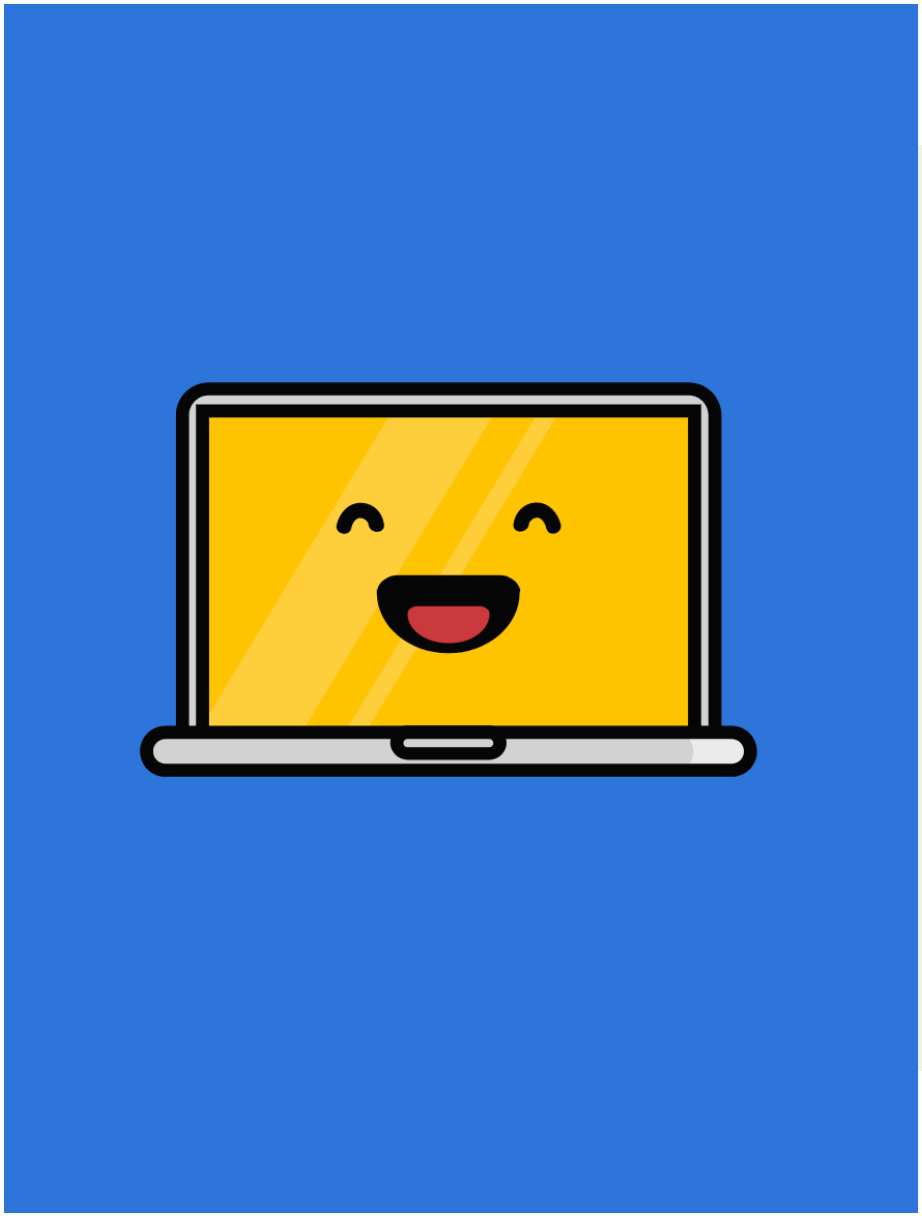


Browser Compatibility

Browser support	CHROME	OPERA	SAFARI	FIREFOX	EDGE
TEMPLATES	STABLE	STABLE	STABLE	STABLE	STABLE
CUSTOM ELEMENTS	STABLE	STABLE	STABLE	POLYFILL DEVELOPING	POLYFILL CONSIDERING
SHADOW DOM	STABLE	STABLE	STABLE	POLYFILL DEVELOPING	POLYFILL CONSIDERING
<SCRIPT TYPE="MODULE">	STABLE	STABLE	10.1	FLAG IN 54	FLAG IN 15
HTML IMPORTS	STABLE	STABLE	POLYFILL ON HOLD	POLYFILL ON HOLD	POLYFILL CONSIDERING

Source: webcomponents.org

Custom Elements Everywhere !



	SCORE
Angular	30/30
CanJS	26/30
Preact	24/30
ReactJS	16/30
Vue	30/30

Courtesy: Rob Dodson Session

Polymer Summit 2017

WC with Angular Demo

Already using Web Components in Production

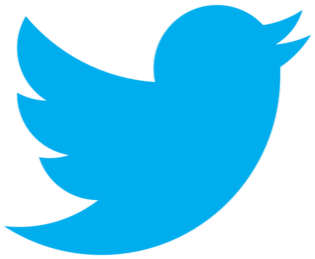
vaadin }>

Bloomberg

<ng-India/>

Using Web Components with Angular

http://bit.ly/WCSuperButton



@mappmechanic

Thank You

E: yehtechnologies@gmail.com