
Web Components with Angular
@Sherrrylst

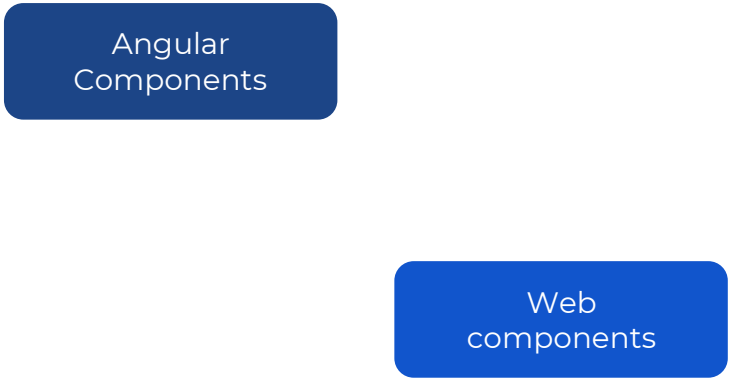
Hello!
I’m Sherry List

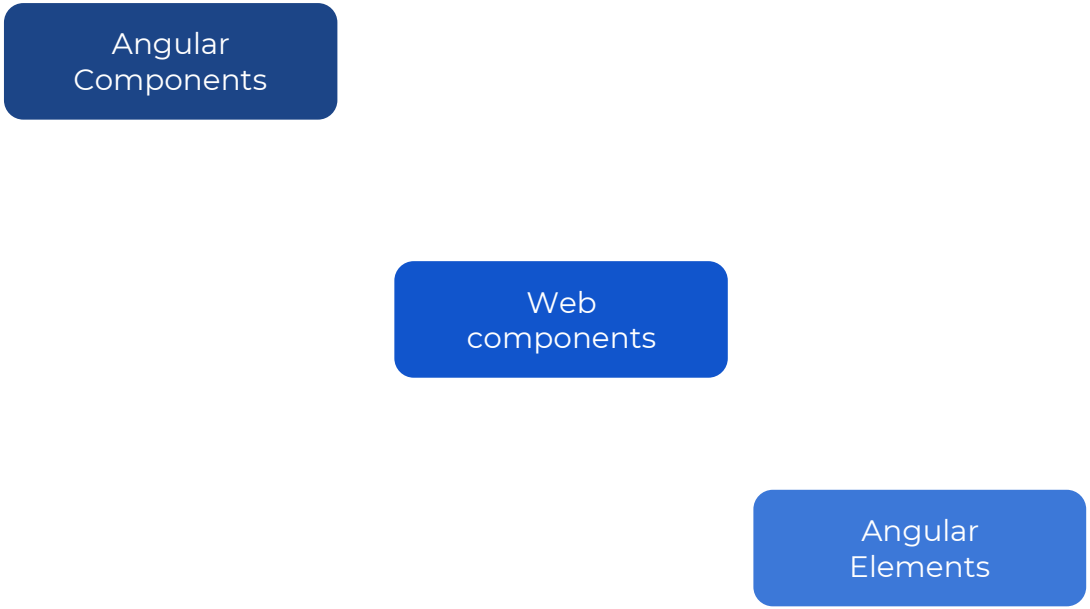
Azure Developer Technical Lead, Microsoft
Women Techmaker Lead

You can find me at @SherryLst

HACK YOUR FUTURE GDG CPH

Angular Components


Angular
Components
Web
components


Angular
Components
Web
components
Angular
Elements

Angular Components

Web components

Angular Elements

What's next?

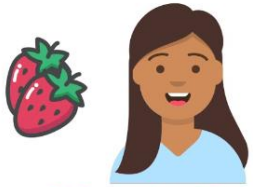

(Angular) Components
@Sherrrylst

search

Home

Products

About



Sherry's Berries

Strawberries

Homegrown and picked daily

€5,00

Blackberries

Homegrown and picked daily

€7,00

Rasberries

Homegrown and picked daily

€8,00

Blueberries

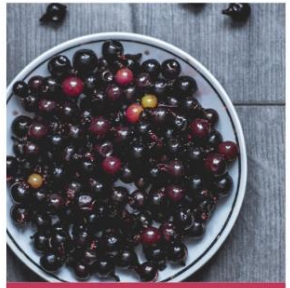
Homegrown and picked daily

€9,00

Cranberries

Homegrown and picked daily

€7,00



Mulberries

Homegrown and picked daily

€12,00

@Sherrrylst

What’s missing:

- Actual **style encapsulation** (nothing leaks in or out without being explicitly allowed)

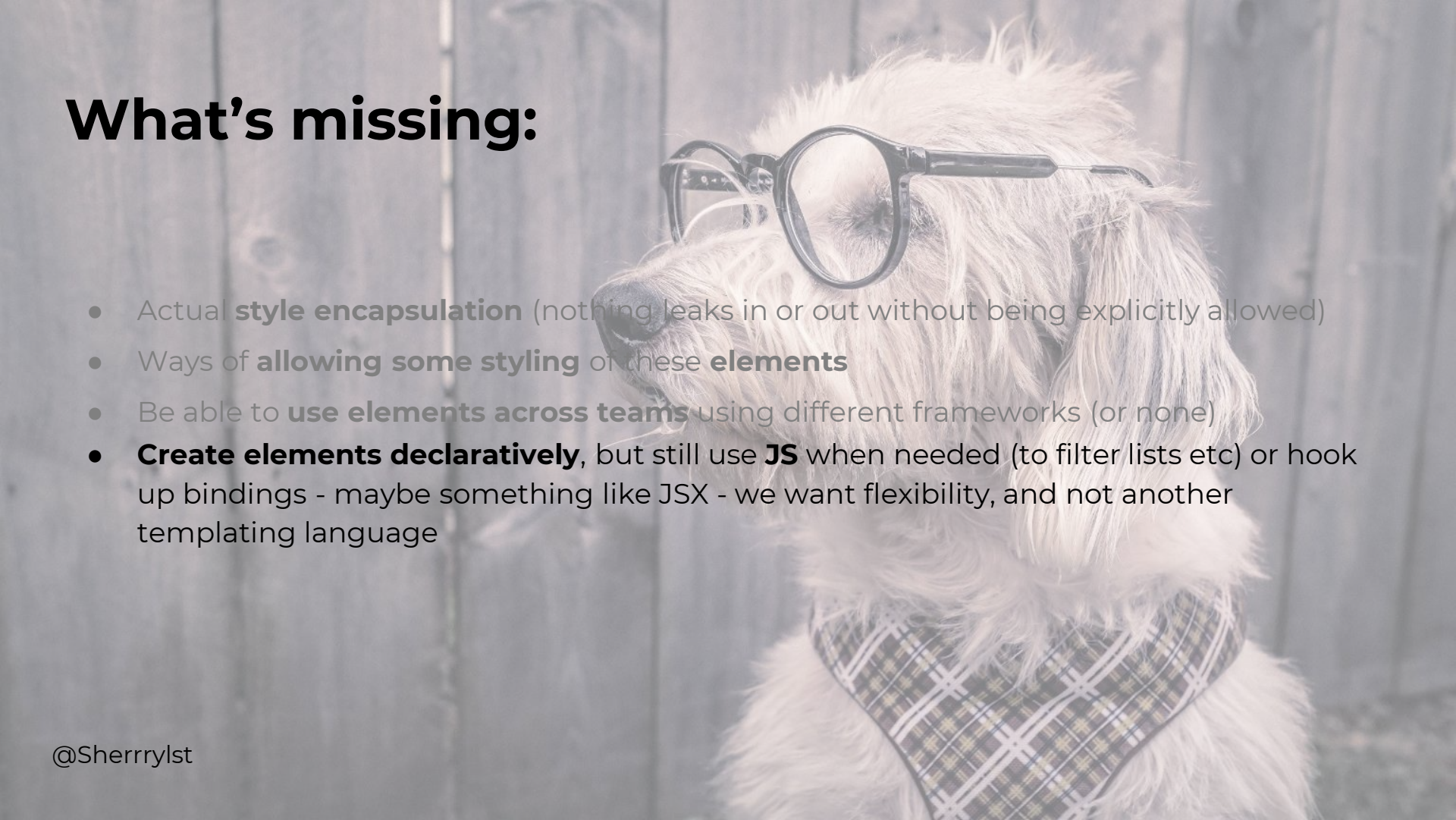
What’s missing:

- Actual **style encapsulation** (nothing leaks in or out without being explicitly allowed)
- Ways of **allowing some styling** of these **elements**

What's missing:

- Actual **style encapsulation** (nothing leaks in or out without being explicitly allowed)
- Ways of **allowing some styling** of these **elements**
- Be able to **use elements across teams** using different frameworks (or none)

What's missing:



- Actual **style encapsulation** (nothing leaks in or out without being explicitly allowed)
- Ways of **allowing some styling** of these **elements**
- Be able to **use elements across teams** using different frameworks (or none)
- **Create elements declaratively**, but still use **JS** when needed (to filter lists etc) or hook up bindings - maybe something like JSX - we want flexibility, and not another templating language


What
does?
@Sherrrylst

Web Components

provides a lot of this

A component that is **platform agnostic**

Their main goal is to **encapsulate** the code for the components into a nice, **reusable** package for maximum **interoperability**.

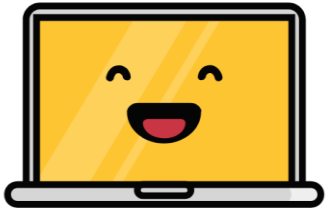
Custom Elements Everywhere

Making sure frameworks and custom elements can be BFFs 

Check out: https://custom-elements-everywhere.com/

@Sherrrylst

Related Issues



Yay! No open issues!

LIBRARY	SCORE	BASIC TESTS	ADVANCED TESTS
Angular 7.2.0	100%	16/16	14/14

@Sherrrylst

Web components consist of three main technologies:

- HTML template
- Custom Elements
- Shadow DOM
- ~~HTML imports~~

Web components consist of three main technologies:

- HTML template

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

StackBlitz

Strawberry history

The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit. The strawberry fruit was mentioned in ancient Roman literature in reference to its medicinal use.

Stawberries from Sherry's garden

Stawberries from Sherry's Berries

@Sherrrylst

HTML Template

StackBlitz

```
<template id="my-shiny-template">
  <div class="template__container">
    <img class="template__image">
    <div class="template__info">
    </div>
  </div>
</template>
```

@Sherrrylst

HTML Template

StackBlitz

```
<div id="image-gallery" class="template__grid">
    <!-- existing image containers will be included here -->
</div>
```

HTML Template

StackBlitz

```
const template = document.querySelector('#red-strawberry-template');
```

HTML Template

StackBlitz

```
const template = document.querySelector('#red-strawberry-template');

// Add the first image
const imgTag = template.content.querySelector("img");
 imgTag.src = "strawberry.jpg";
 imgTag.alt = "Strawberries on the table";
```

@Sherrrylst

HTML Template

StackBlitz

```
const template = document.querySelector('#red-strawberry-template');

// Add the first image
const imgTag = template.content.querySelector("img");
 imgTag.src = "strawberry.jpg";
 imgTag.alt = "Strawberries on the table";

// Add the first content
const info = template.content.querySelector("span");
info.textContent = "Stawberries from Sherry's garden";
```

HTML Template

StackBlitz

```
// Clone the new gallery and insert the DOM
const imageGallery = document.querySelector("#image-gallery");
```

HTML Template

StackBlitz

```
// Clone the new gallery and insert the DOM
const imageGallery = document.querySelector("#image-gallery");
const clone = document.importNode(template.content, true);
```

@Sherrylst

HTML Template

StackBlitz

```
// Clone the new gallery and insert the DOM
const imageGallery = document.querySelector("#image-gallery");
const clone = document.importNode(template.content, true);
imageGallery.appendChild(clone);
```

@Sherrrylst

Why HTML templates?

- Browser parse it once
- Fast
- Easy to use

Web components consist of three main technologies:

- HTML Template
- Custom Elements

Naming your custom element

<red-button></red-button>

<**my-unique-wc**></**my-unique-wc**>

Naming the attributes

```
<my-unique-wc
    title="I am a Title"
    xyz="No idea what I am!">
</my-unique-wc>
```

Custom elements lifecycle

Native	Angular	React	Vue
constructor	ngOnInit	Constructor	Creates
connectedCallback	ngAfterContentChecked	componentDidMount	mounted
disconnectedCallback	ngOnDestroy	componentWillUnmount	destroy
attributeChangedCallback	ngOnChanges	componentDidUpdate	updated

Source: https://youtu.be/j-5uodRMW_Q

Custom elements

```
<bb-red-strawberry
    img="strawberry.jpg"
    description="Strawberries from Sherry's Garden">
</bb-red-strawberry>
```

Custom elements

```
class BBRedStrawberryElement extends HTMLElement {
  constructor() {
    super();
  }
}
```

Custom elements

```
class BBRedStrawberryElement extends HTMLElement {
    constructor() {
      super();
    }
 }

// Define custom element
customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

Custom elements

```
connectedCallback() {
    this.innerHTML = template;
    this._$image = this.querySelector("#element-image");
    this._$description = this.querySelector("#element-description");
    this._render(this);
}
```

Custom elements

```
_render({ img, description }) {
    description = description || "Description is missing";
    this._$image.alt = description;
    this._$image.src = img || "missing-image.jpg";

    const figcaption = document.createElement("figcaption");
    figcaption.textContent = description;
    figcaption["aria-label"] = "product name";
    this._$description.appendChild(figcaption);
}
```

Custom elements

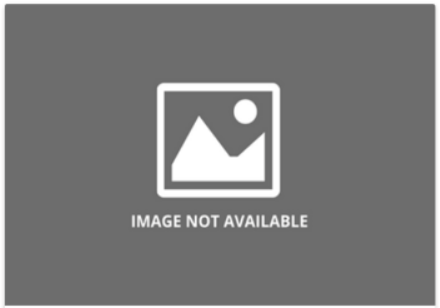
```
...
static get observedAttributes() {
return ["description", "img"];
}

attributeChangedCallback(name, oldValue, newValue) {
this[name] = newValue;
}

...
```

Strawberry history

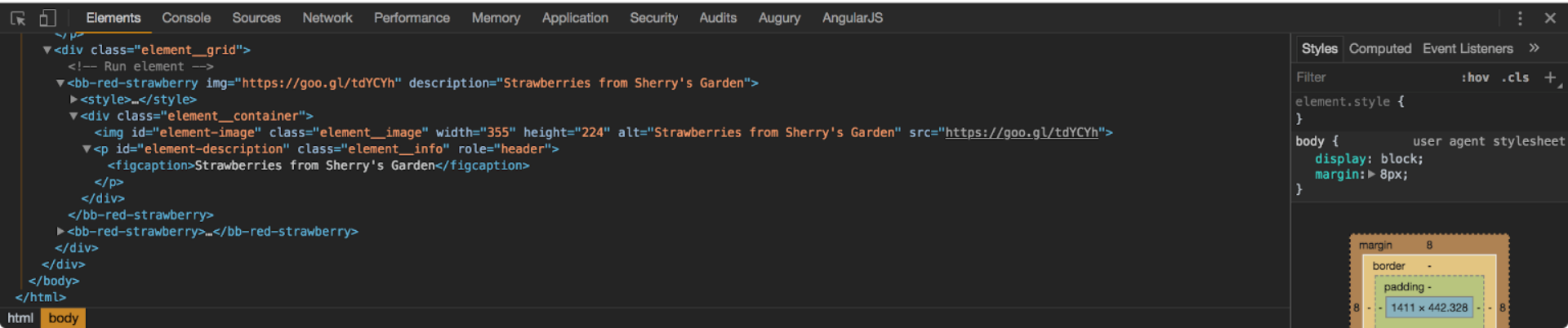
StackBlitz

The first garden strawberry was grown in Brittany, France during the late 18th century. ßPrior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit.

Strawberries from Sherry's Garden



Description is missing




@Sherrrylst

Global CSS

```
<style>

 p {
        font-family: 'Lato', sans-serif;
        font-size: 0.9em;
        max-width: 760px;
        line-height: 1.6em;
        color: red;
 }
```

Strawberry history

The first garden strawberry was grown in Brittany, France during the late 18th century. ßPrior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit.

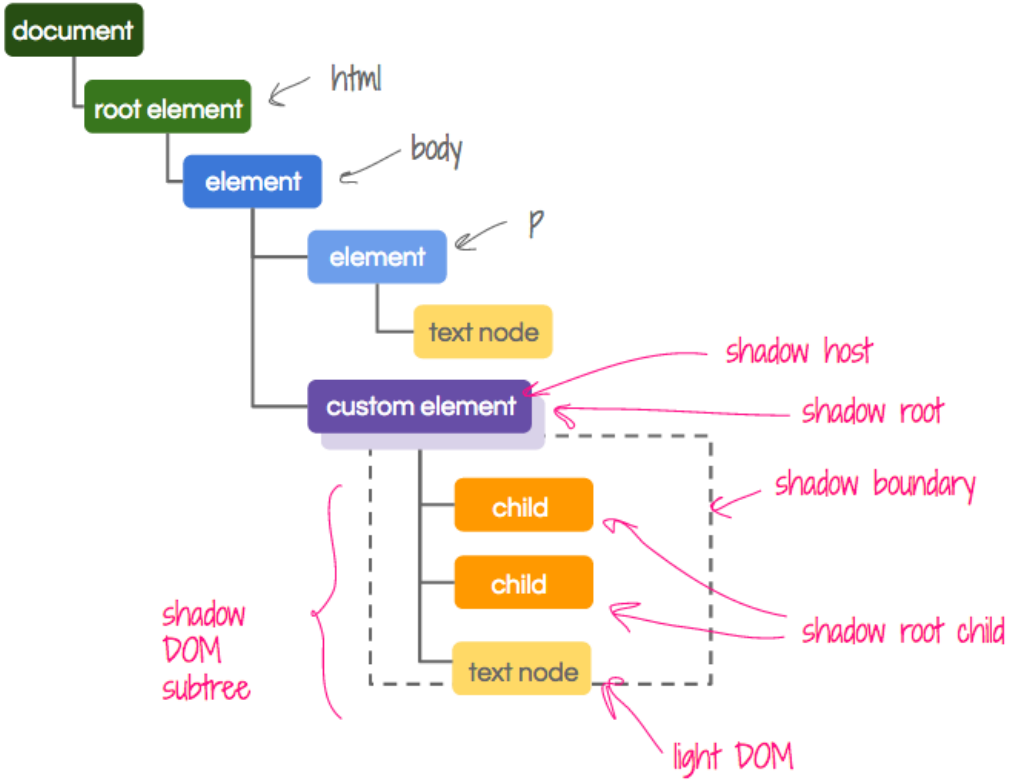
Strawberries from Sherry's Garden

Description is missing

Web components consist of three main technologies:

- HTML Template
- Custom Elements
- Shadow DOM

Shadow DOM



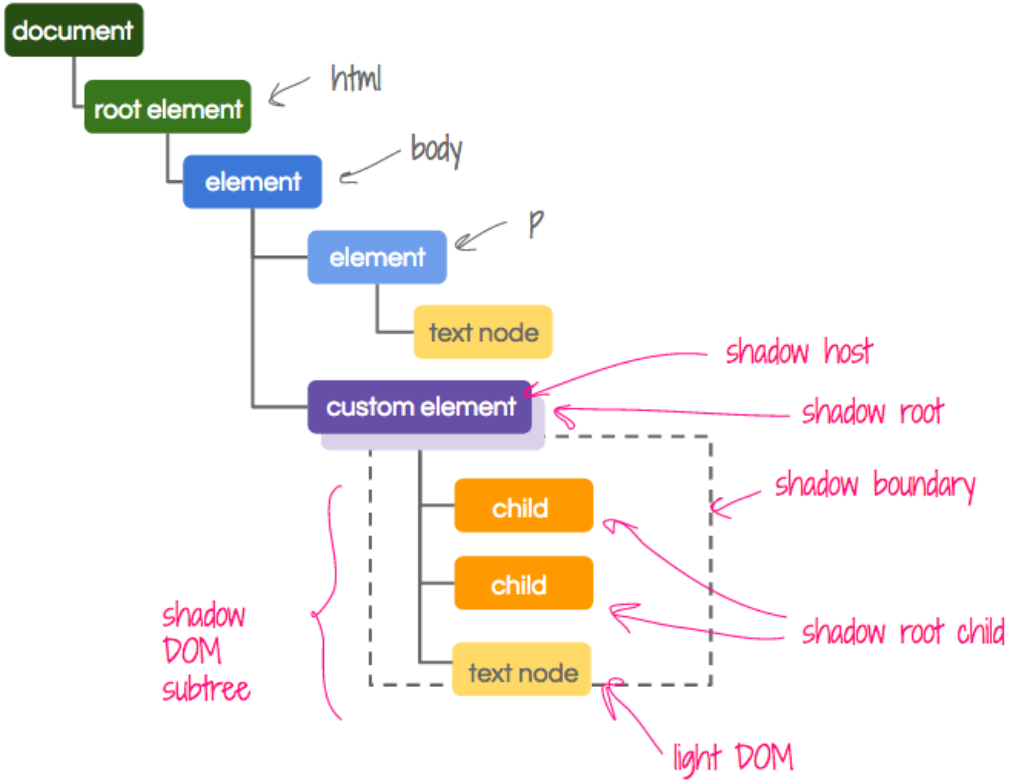
https://girliemac.com/presentation-slides/webcomponents/images/dom-tree.png

Shadow DOM

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  // Shadow DOM
  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
 }
```


Slots
@Sherrrylst

Shadow DOM



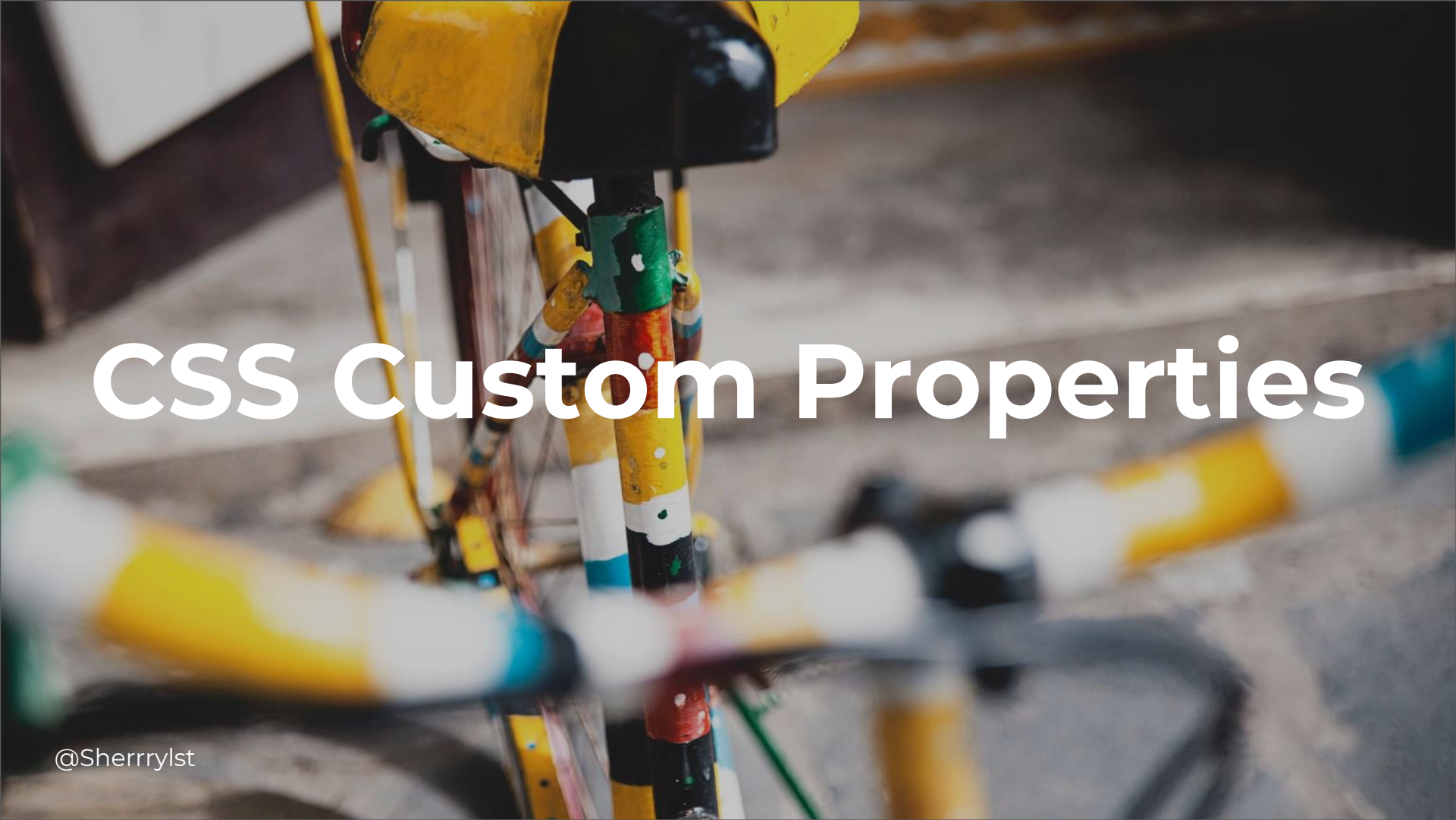
@Sherrrylst

Template

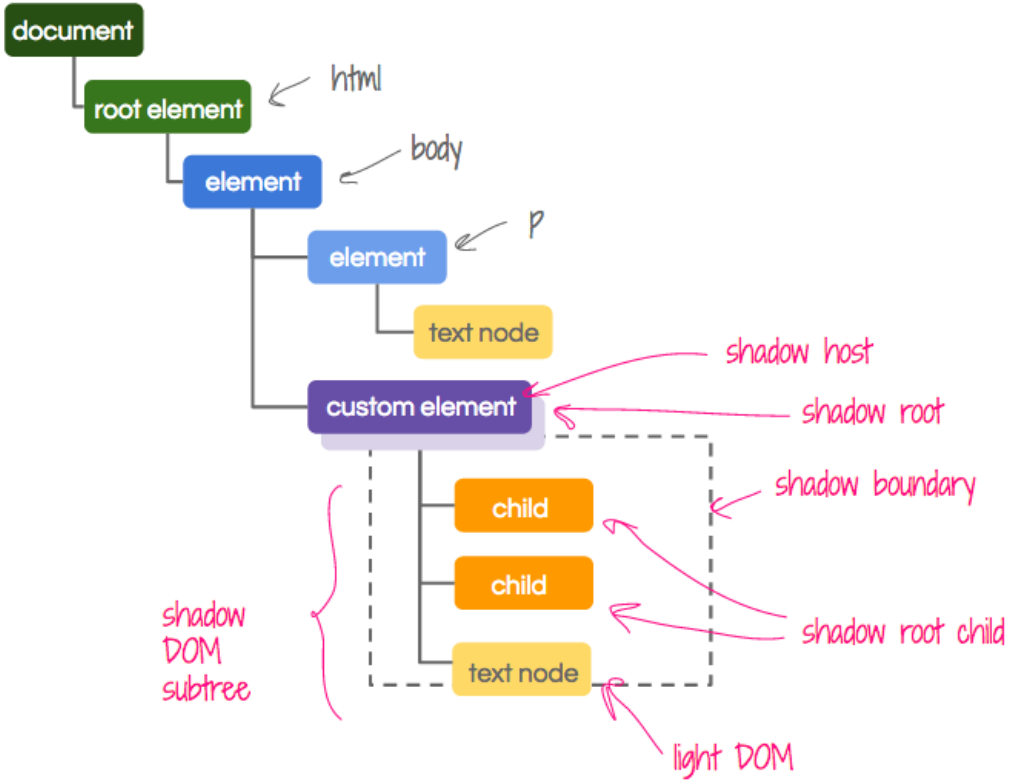
```
...
    <img id="shadow-image" class="shadow__image">
    <div id="shadow-info" class="shadow__info">
      <slot name="title" id="title" role="header" class="shadow__title"></slot>
      <slot name="description" id="description" class="shadow__description"></slot>
    </div>
...
```

Custom Element

```
<bb-red-strawberry
    img="strawberry.jpg">
     <div slot="title"><i>Strawberries</i></div>
     <div slot="description">Sherry's berries finest strawberries</div>
 </bb-red-strawberry>
```


CSS Custom Properties
@Sherrrylst

Shadow DOM



@Sherrrylst

Template

```
<style>
    .container {
        width: 320px;
        height: 280px;
        background-color: var(--background-color, #fff);
        }
<style>
```

CSS custom properties

```
<bb-red-strawberry
    img="strawberry.jpg"
    style="--background-color: #A11B38;">
    <div slot="title"><i>Strawberries</i></div>
    <div slot="description">Sherry's berries finest strawberries</div>
 </bb-red-strawberry>
```


Remember:
Documentation
@Sherrrylst


:host
@Sherrrylst

Template

```
<style>
    :host(bb-red-strawberry) {
        /* Applies if the host is a <bb-red-strawberry> element.*/
        font-weight: bold;
        width: 320px;
        height: 280px;
        background-color: var(--background-color, #fff);
        box-shadow: 0 2px 2px 0 rgba (0, 0, 0, .14);
        border-radius: 2px;
    }
<style>
```

@Sherrrylst

Strawberry history

StackBlitz

The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit.

Strawberries
Sherry's berries finest strawberries

IMAGE NOT AVAILABLE

Description is missing

Elements Console Sources Network Performance Memory Application Security Audits Augury AngularJS

```
<div class="element__grid">
  <!-- Run element -->
  <bb-red-strawberry img="https://goo.gl/tdYCYh" style="--background-color: #A11B38; --color:white">
    #shadow-root (open)
      <style>…</style>
      <img id="shadow-image" class="shadow__image" alt="Description is missing" src="https://goo.gl/tdYCYh"> == $0
      <div id="shadow-info" class="shadow__info">…</div>
    <div slot="title">Strawberries</div>
    <div slot="description">Sherry's berries finest strawberries</div>
  </bb-red-strawberry>
  <!-- ADD image and description -->
  <bb-red-strawberry>…</bb-red-strawberry>
 </div>
</div>
</body>
</html>
```

html body div.main__container div.element__grid bb-red-strawberry #shadow-root img#shadow-image.shadow__image

Styles Computed Event Listeners

Filter :hov .cls

```
element.style {
}
.shadow__image {                <style>…</style>
  width: 320px;
  height: 220px;
  contain: cover;
  border-top-left-radius: 2px;
  border-top-right-radius: 2px;
}
```

Inherited from #shadow-root (open)

```
Style Attribute {
  --background-color: #A11B38;
  --color: white;
}
```


@Sherrrylst

Global CSS

```
<style>

 p {
        font-family: 'Lato', sans-serif;
        font-size: 0.9em;
        max-width: 760px;
        line-height: 1.6em;
        color: red;
 }
```

Strawberry history

StackBlitz

The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit.

Strawberries
Sherry's berries finest strawberries

IMAGE NOT AVAILABLE

Description is missing

Elements Console Sources Network Performance Memory Application Security Audits AngularJS Augury

```
<p>
  "The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries
  and cultivated selections from wild strawberry species were the common source of the fruit."
</p>
<div class="element__grid">
  <!-- Run element -->
  <bb-red-strawberry img="https://goo.gl/tdYCYh" style="--background-color: #A11B38; --color:white">
    #shadow-root (open)
    <div slot="title">Strawberries</div>
    <div slot="description">Sherry's berries finest strawberries</div>
  </bb-red-strawberry>
  <!-- ADD image and description -->
  <bb-red-strawberry>…</bb-red-strawberry>
</div>
</div>
</body>
</html>
```

html body

Styles Computed Event Listeners »

Filter :hov .cls +

```
element.style {
}
body {                    user agent stylesheet
  display: block;
  margin: 8px;
}
```

margin 8
border -
padding -
8 - - 1411 × 443.938 - - 8


We made it!
@Sherrrylst

@Sherrrylst

Angular Elements

@Sherrrylst

terminal

```
npm i -g @angular/cli
```

terminal

```
npm i -g @angular/cli
```

```
ng new bb-card --prefix bb --inline-template --style=scss
```

terminal

```
npm i -g @angular/cli

ng new bb-card --prefix bb --inline-template --style=scss

cd bb-card
```

terminal

```
npm i -g @angular/cli

ng new bb-card --prefix bb --inline-template --style=scss

cd bb-card

ng add @angular/elements
```

terminal

```
npm i -g @angular/cli

ng new bb-card --prefix bb --inline-template --style=scss

cd bb-card

ng add @angular/elements

npm install @webcomponents/custom-elements --save
```

polyfills.ts

```
import '@webcomponents/custom-elements/custom-elements.min';
```

tsconfig.json

```
{
    "compileOnSave": false,
    "compilerOptions": {
        ...
        "target": "es2015",
        ....
        }
    }
}
```

terminal

```
ng generate component card
```

card.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
   selector: 'bb-card',
   template: ` ... `,
   styleUrls: ['./bb-card.scss'],
})

export class CardComponent implements OnInit {
 constructor() { }
 ngOnInit() {}
}
```


Look back...
@Sherrrylst

Web component (Native)

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
 }

customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

Web component (Native)

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
 }

customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

ViewEncapsulation.ShadowDom

This encapsulation mode uses the Shadow DOM to **scope styles only to this specific component**.

card.component.ts

```
import { Component, OnInit, ViewEncapsulation, Input } from
'@angular/core';

@Component({
   selector: 'bb-card',
   template: ` ... `,
   styleUrls: ['./bb-card.scss'],
   encapsulation: ViewEncapsulation.ShadowDom
})
```

Make our code **Angular** friendly

card.component.ts

```
template: `
<img id='shadow-image' class='shadow__image' src='{{ img }}' alt='title'>
<div id='shadow-info' class='shadow__info'>
  <h1 name='title' id='title' role='header' class='shadow__title'>{{ title }}</h1>
  <p name='description' id='description' class='shadow__description'>{{ description }}</p>
</div>
`
```

card.component.ts

```
export class CardComponent implements OnInit{
  @Input() title?: string = 'default title';
  @Input() description?: string = 'default description';
  @Input() img?: string = 'undefined.png';

  constructor() { }
  ngOnInit() {}
}
```

card.component.scss

```
:host {
 display: block;
 width: 320px;
 height: 300px;
 background-color: var(--background-color, #fff);
[...]
   .shadow__image {
     [...]
   }
   .shadow__info {
    [...]
     }
}
```

Register our **component** in NgModule

app.module.ts

```
import { NgModule, Injector } from '@angular/core';
import { createCustomElement } from '@angular/elements';

@NgModule({
   declarations: [ CardComponent ],
   imports: [ BrowserModule ],
    entryComponents: [ CardComponent ],
})
```

Look back...

Web component (Native)

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
 }

customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

Web component (Native)

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
 }
// Define custom element
customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

app.module.ts

```
@NgModule({
  declarations: [CardComponent],
  imports: [BrowserModule],
  entryComponents: [CardComponent],
})

export class AppModule {
  constructor(private injector: Injector) {
    const bbCard = createCustomElement(CardComponent, { injector });
    customElements.define('bb-card', bbCard);
  }
```

app.module.ts

```
@NgModule({
   declarations: [CardComponent],
   imports: [BrowserModule],
   entryComponents: [CardComponent],
})

export class AppModule {
  constructor(private injector: Injector) {
    const bbCard = createCustomElement(CardComponent, { injector });
    customElements.define('bb-card', bbCard);
  }
  ngDoBootstrap() {}
}
```

terminal

```
ng build
```

Blackberries

Fresh from garden

@Sherrrylst

```
<!doctype html>
<html lang="en">
 ▸<head>…</head>
… ▾<body> == $0
   ▾<bb-card title="Blackberries" description="Fresh from garden" src=
   "https://goo.gl/zfhMKP" ng-version="6.1.0">
    ▾#shadow-root (open)
     ▸<style>…</style>
      <img alt="title" class="shadow__image" id="shadow-image" src=
      "https://goo.gl/zfhMKP">
     ▾<div class="shadow__info" id="shadow-info">
       <h1 class="shadow__title" id="title" name="title">Blackberries
       </h1>
       <p class="shadow__description" id="description" name=
       "description">Fresh from garden</p>
      </div>
   </bb-card>
   <script type="text/javascript" src="runtime.js"></script>
   <script type="text/javascript" src="polyfills.js"></script>
   <script type="text/javascript" src="styles.js"></script>
   <script type="text/javascript" src="scripts.js"></script>
   <script type="text/javascript" src="vendor.js"></script>
   <script type="text/javascript" src="main.js"></script>
  </body>
</html>
```

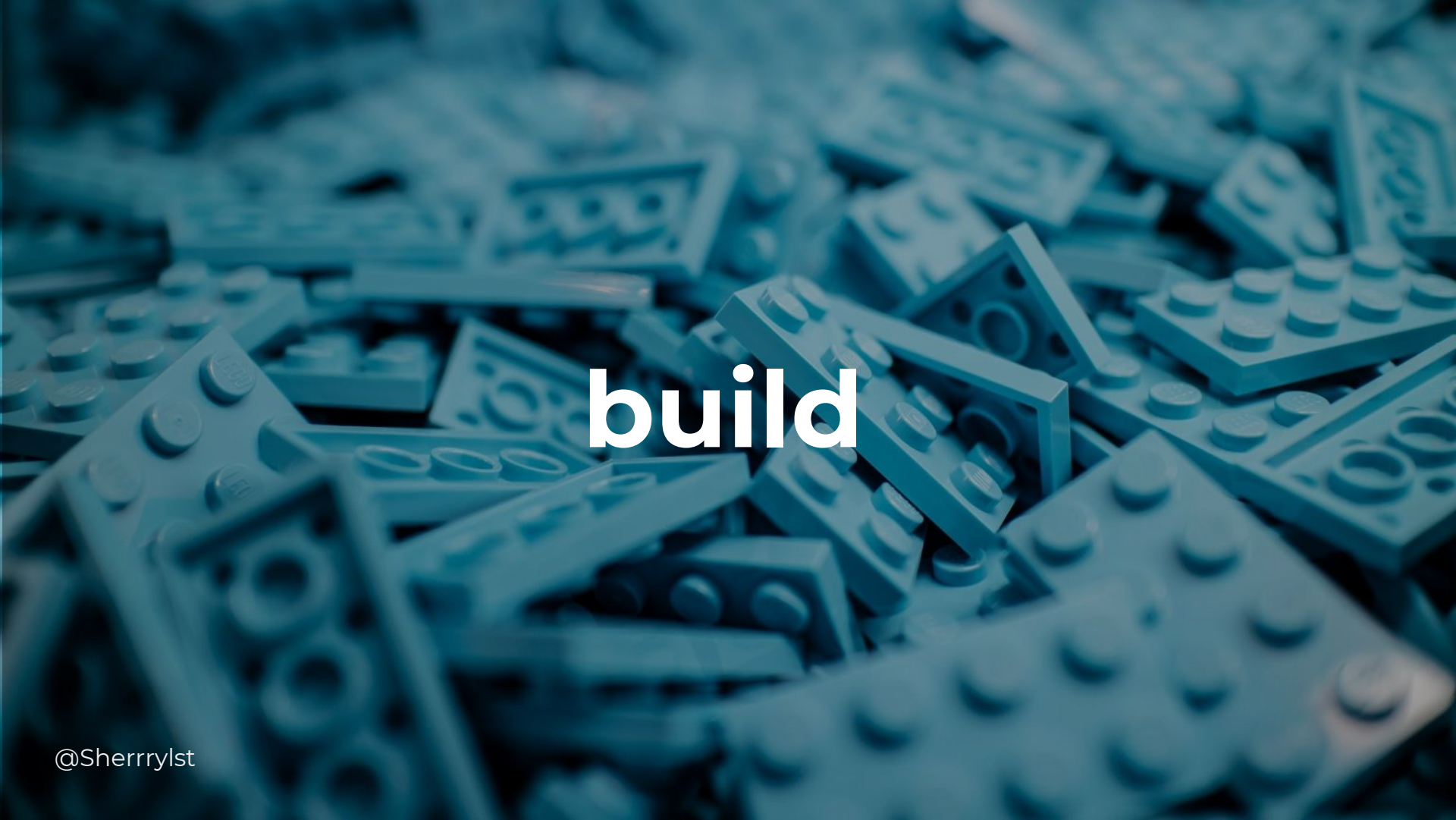
```
element.style {
}
body {                    user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```

margin 8 · border - · padding - · 682 × 300 · 8 8 8 8

html body


@Sherrrylst

Wait a min



@Sherrrylst

It’s not that easy!

- Huge bundle size
- No support for one bundle
- -> Eject
- Complicated!

ngx-build-plus

@ManfredSteyer

Check out: https://github.com/manfredsteyer/ngx-build-plus

@Sherrylst

It’s #awesome!

- Extends Cli
- No Eject
- Build a single bundle
- No need to add Angular multiple times to your project
- Universal module

terminal

```
ng add ngx-build-plus
```

angular.json

```
[...]
"architect": {
   "build": {
         "builder": "ngx-build-plus:build",
      [...]
   }
}
[...]
```

webpack.extra.js

```
module.exports = {
    "externals": {
        "rxjs": "rxjs",
        "@angular/core": "ng.core",
        "@angular/common": "ng.common",
        "@angular/platform-browser": "ng.platformBrowser",
        "@angular/elements": "ng.elements"
    }
}
```

terminal

```
ng build  --prod
          --extraWebpackConfig webpack.extra.js
          --output-hashing none
          --single-bundle true
```

Now our component is **ready**!

Ivy

The new backwards-compatible Angular renderer:

- Speed improvements
- Bundle size reduction
- Increasing flexibility

How do we **use** it?

terminal

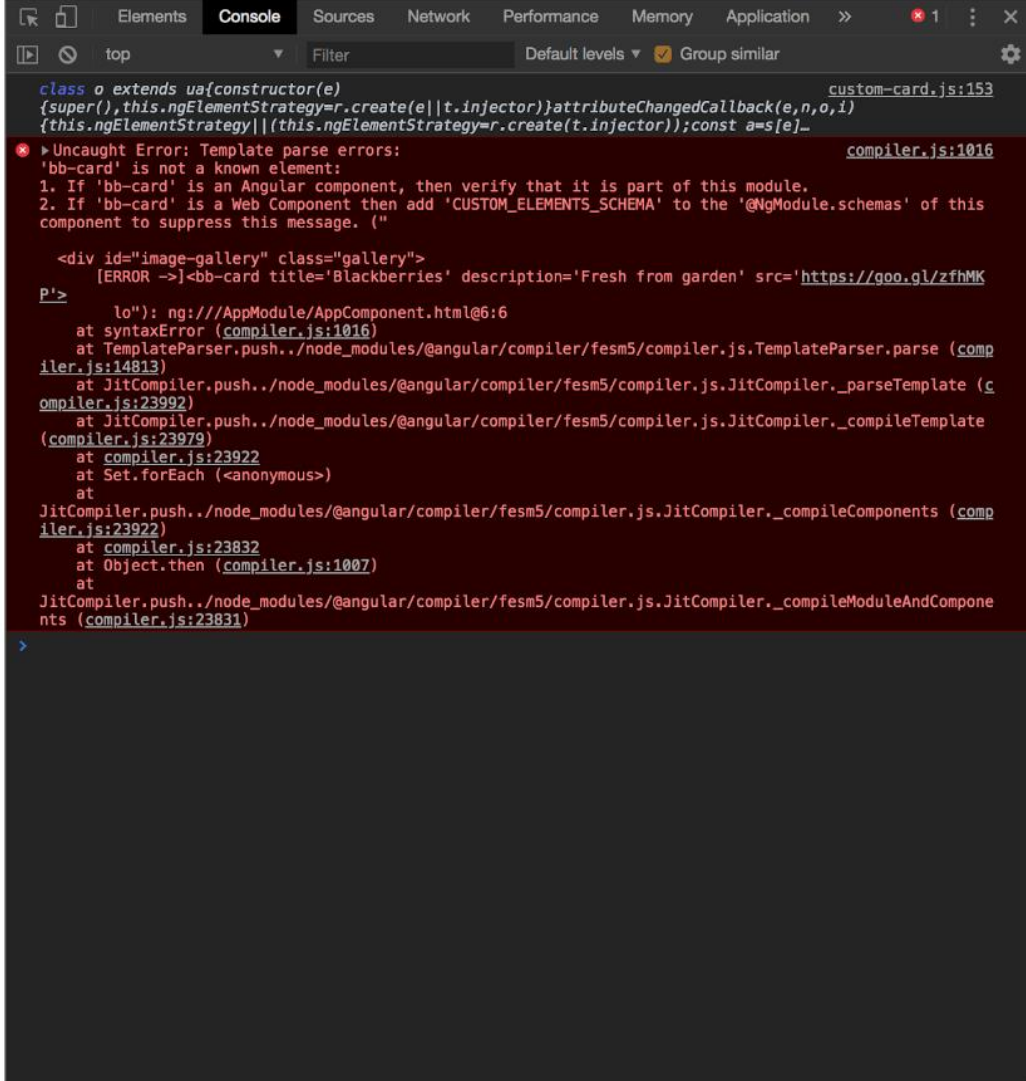
```
npm install @webcomponents/custom-elements --save
```

polyfills.ts

```
import '@webcomponents/custom-elements/custom-elements.min';
```

app.component.ts

```
import { Component } from '@angular/core';
import * as bbCard from '../web-components/bb-card/bb-card.js';
@Component({
  selector: 'berry-root',
  template: `
      <bb-card
          title='Strawberries'
          description='Fresh from Sherry garden' img='../assets/images/strawberry-
          unx.jpeg' style="--background-color: #A11B38; --color: #ffffff;">
          loading...
      </bb-card>
`,
  styleUrls: ['./app.component.css']
})
export class AppComponent {...}
```


Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
Group similar
class o extends ua{constructor(e) {super(),this.ngElementStrategy=r.create(e||t.injector)}attributeChangedCallback(e,n,o,i) {this.ngElementStrategy||(this.ngElementStrategy=r.create(t.injector));const a=s[e]…
custom-card.js:153
Uncaught Error: Template parse errors:
'bb-card' is not a known element:
1. If 'bb-card' is an Angular component, then verify that it is part of this module.
2. If 'bb-card' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message. ("

[ERROR ->]<bb-card title='Blackberries' description='Fresh from garden' src='https://goo.gl/zfhMKP'>
lo"): ng:///AppModule/AppComponent.html@6:6
at syntaxError (compiler.js:1016)
at TemplateParser.push../node_modules/@angular/compiler/fesm5/compiler.js.TemplateParser.parse (compiler.js:14813)
at JitCompiler.push../node_modules/@angular/compiler/fesm5/compiler.js.JitCompiler._parseTemplate (compiler.js:23992)
at JitCompiler.push../node_modules/@angular/compiler/fesm5/compiler.js.JitCompiler._compileTemplate (compiler.js:23979)
at compiler.js:23922
at Set.forEach (<anonymous>)
at JitCompiler.push../node_modules/@angular/compiler/fesm5/compiler.js.JitCompiler._compileComponents (compiler.js:23922)
at compiler.js:23832
at Object.then (compiler.js:1007)
at JitCompiler.push../node_modules/@angular/compiler/fesm5/compiler.js.JitCompiler._compileModuleAndComponents (compiler.js:23831)
compiler.js:1016

@Sherrrylst

app.module.ts

```
@NgModule({
  [...]
  schemas: [
    CUSTOM_ELEMENTS_SCHEMA
  ]
})

export class AppModule {
}
```

Strawberry history

The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit. The strawberry fruit was mentioned in ancient Roman literature in reference to its medicinal use.

Strawberries

Fresh from Sherry garden

Strawberries

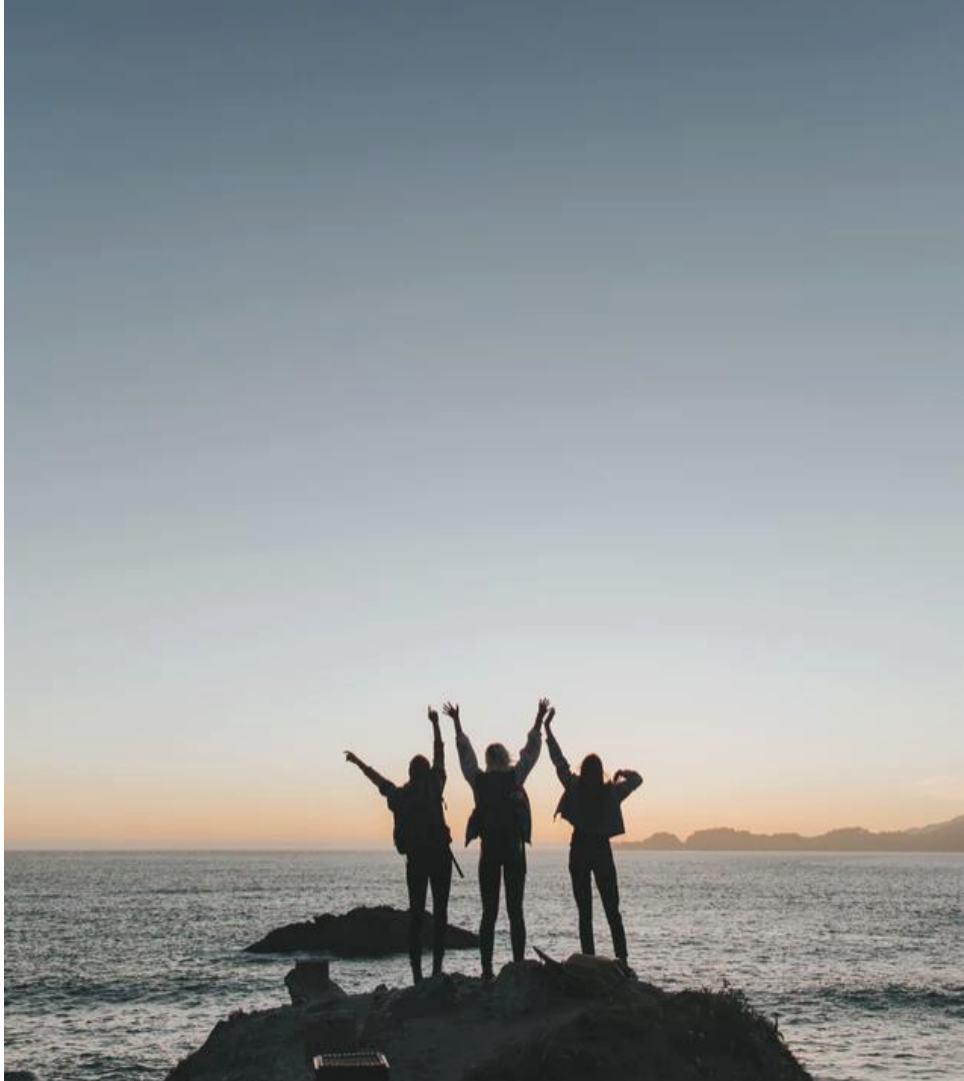
Fresh from Ana garden

@Sherrrylst

We are almost done!

- ✅ Web components fundamentals
- ✅ Create a web component
- ✅ Create a web component with Angular
- ✅ How to build an Angular element
- ✅ Add a web component to an existing Angular project

@Sherrrylst



Web Components rock?

@Sherrylst

Yes they DO!

- Maximum interoperability
- Support from Frameworks (Tooling)
- Many success stories
- Browsers are ready

Browser support	CHROME	OPERA	SAFARI	FIREFOX	EDGE
HTML TEMPLATES	STABLE	STABLE	STABLE	STABLE	STABLE
CUSTOM ELEMENTS	STABLE	STABLE	STABLE	STABLE	POLYFILL DEVELOPING
SHADOW DOM	STABLE	STABLE	STABLE	STABLE	POLYFILL DEVELOPING
ES MODULES	STABLE	STABLE	STABLE	STABLE	STABLE

@Sherrylst

*https://webcomponents.org

What’s next?

@Sherrrylst

lit-html

Check out: https://youtu.be/lo6JjgckHbg

@Sherrylst

What is lit-html?

- Small library
- <template> in JS with template literal
- Extremely fast (Partial update)
- Customizable and extensible (Directives)

Status: In Development

@Sherrrylst

lit-html

```
import { render, html } from 'lit-html';

class BbRedStrawberryElement extends HTMLElement {
    constructor() {
        super();
        const template = document.createElement("template");
        this.attachShadow({ "mode": "open" });
        this.shadowRoot.appendChild(template.content.cloneNode(true));
    }
}
```

lit-html

```
connectedCallback() {
    render(this._render(this), this.shadowRoot);
}
```

lit-html

```
connectedCallback() {
        render(this._render(this), this.shadowRoot);
}

// Defines element markup
_render({ img, title, description, productPrice, color }) {
        return html`
```

lit-html

```
<div class="main-container">
    <div id="photo">
        <img src=${ img || "missing.jpg" } alt=${ title || 'N/A' }></img>
    </div>
    <div id="info">
        <h3 id="title" aria-label="product title">${ title || 'N/A' }</h3>
            ${ description ? html`<p id="description" aria-label="product description">${ description || 'N/A' }
            </p>` : '' }
            ${ productPrice ? html` <p id="price" aria-label="product price">${ productPrice } / kilo</p>` : '' }
    </div>
</div>
```

Strawberry history

StackBlitz

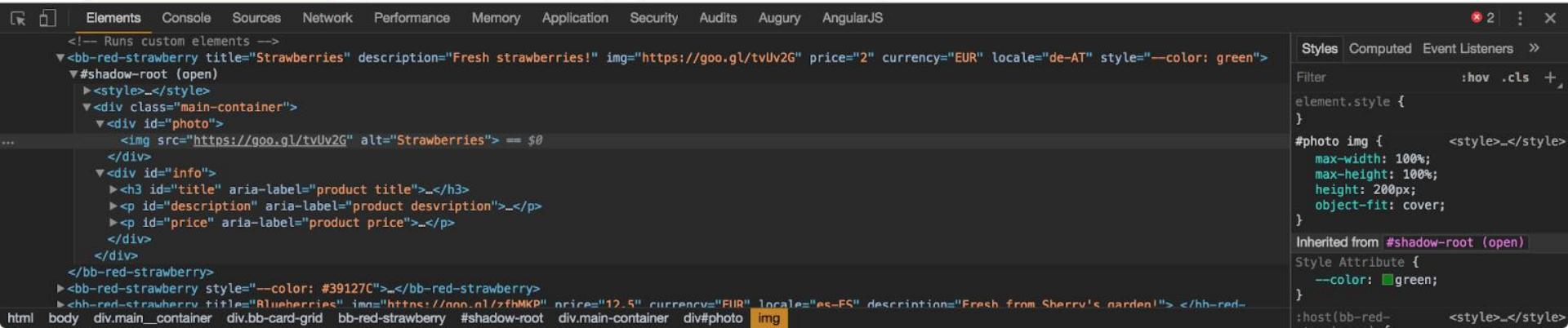
The first garden strawberry was grown in Brittany, France during the late 18th century. Prior to this, wild strawberries and cultivated selections from wild strawberry species were the common source of the fruit. The strawberry fruit was mentioned in ancient Roman literature in reference to its medicinal use.

Strawberries
Fresh strawberries!
€ 2,00 / kilo

IMAGE NOT AVAILABLE

N/A

Blueberries
Fresh from Sherry's garden!
12,50 € / kilo



lit-element

Check out: https://youtu.be/ypPRdtjGooc

@Sherrylst

What is lit-element?

- Base class for creating web components
- Uses lit-html to render into the element's Shadow DOM
- React to changes
- Extremely fast & light

Status: In Development

lit-element

```
import { litElement, html } from 'lit-html';
class BbRedStrawberryElement extends litElement {
static get properties() {
      return {
        title: { type: String },
        description: { type: String },
        currency: { type: String },
        price: { type: String },
        img: { type: String }
      }
    }
}
```

@Sherrylst

lit-element

```
render() {
    return html`
        <div class="main-container">
            <div id="photo">
                <img src=${ img || "missing.jpg" } alt=${ title || 'N/A' }></img>
            </div>
            <div id="info">
                <h3 id="title" aria-label="product title">${ title || 'N/A' }</h3>
                ${ description ? html`<p id="description" aria-label="product description">
                ${ description || 'N/A' }
                </p>` : '' }
                ${ productPrice ? html` <p id="price" aria-label="product price">
                ${ productPrice } / kilo</p>` : '' }
            </div>
        </div> `
}
```

```
class DemoElement extends LitElement {

  @property()
  filename;

  render() {
    return html`
      <p>
        ${fetchContent(this.filename,
          (content) => html`content: ${content}`,
          () => html`Loading...`,
          () => html`Please enter a filename`,
          (e) => html`Error: ${e.message}
        )}
      </p>
    `;
  }
}
```


Check out: https://youtu.be/ypPRdtjGooc

CSS Shadow Parts

::part()

Check out: https://meowni.ca/posts/part-theme-explainer/

@Sherrrylst

::part

```
<x-foo>
  #shadow-root
  <div part="some-box"><span>...</span></div>
  <input part="some-input">
   <div>...</div> /* not styleable
</x-foo>
```

Check out: https://meowni.ca/posts/part-theme-explainer/

@Sherrylst

::part

At a document which has <x-foo>:

```
x-foo::part(some-box) { … }

x-foo::part(some-box):hover { … }

x-foo::part(some-input)::placeholder { … }
```

Check out: https://meowni.ca/posts/part-theme-explainer/

@Sherrylst

Web Components do ROCK!

Angular  Web Components

Sherry List
@SherrryLst

Ana Cidre
@AnaCidre_

Azure for JavaScript developers

https://aka.ms/AA413rz

Best practices

- https://w3ctag.github.io/webcomponents-design-guidelines/
- https://github.com/webcomponents/gold-standard/wiki
- https://developers.google.com/web/fundamentals/web-components/best-practices

Sources

- https://www.softwarearchitekt.at/post/2018/07/13/angular-elements-part-i-a-dynamic-dashboard-in-four-steps-with-web-components.aspx
- https://meowni.ca/posts/part-theme-explainer/
- https://medium.com/google-developer-experts/are-web-components-a-thing-5a116b1da7e4
- https://www.telerik.com/blogs/web-components-101-an-introduction-to-web-components
- https://www.softwarearchitekt.at/post/2019/01/27/building-angular-elements-with-the-cli.aspx