Global CSS Scope – Circumvent problems

- INTRO
- BEM
- CSS MODULES
- CSS IN JS
- SHADOW DOM
- ANGULAR VIEW ENCAPSULATION
- APPROX. 30 MIN – HOPEFULLY!!

ALEXANDER SILBERSCHNEIDER
@SILBERXANDER

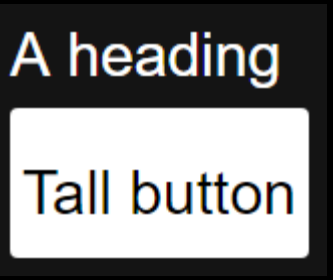
Global CSS Scope

CIRCUMVENT ISSUES

Base selector side effects

```
<h1>A heading</h1>

<button>
  <h1>
    Tall button
  </h1>
</button>
```

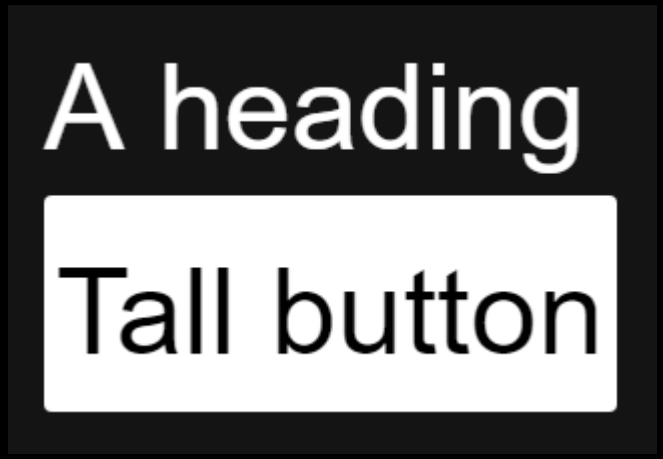
```
h1 {
  font-size: 24px;
}
```



Base selector side effects - oops

```
<h1>A heading</h1>

<button>
  <h1>
    Tall button
  </h1>
</button>
```

```
h1 {
  font-size: 36px;
}
```



Base selector side effects – “solution”

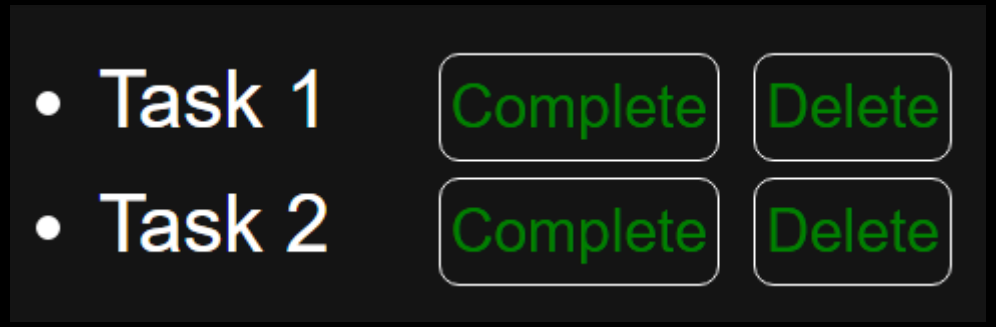
```
<h1>A heading</h1>

<button>
  <h1>
    Tall button
  </h1>
</button>
```

```
h1 {
  font-size: 36px;
}

button h1  {
  font-size: 24px;
}
```

Match on child elements

```
<div class="tasks">
  <ul class="task-items">
    <li>
      Task 1
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
    <li>
      Task 2
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
  </ul>
</div>
```

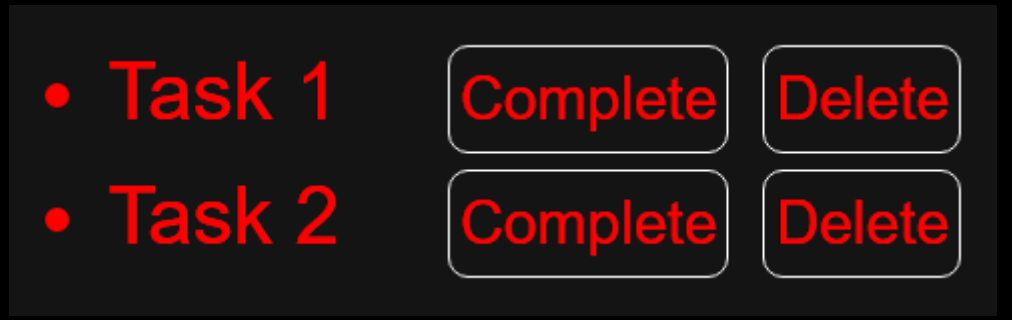
```
.actions li {
    color: green;
}
```



Red task list items – match on children

```
<div class="tasks">
  <ul class="task-items">
    <li>
      Task 1
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
    <li>
      Task 2
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
  </ul>
</div>
```

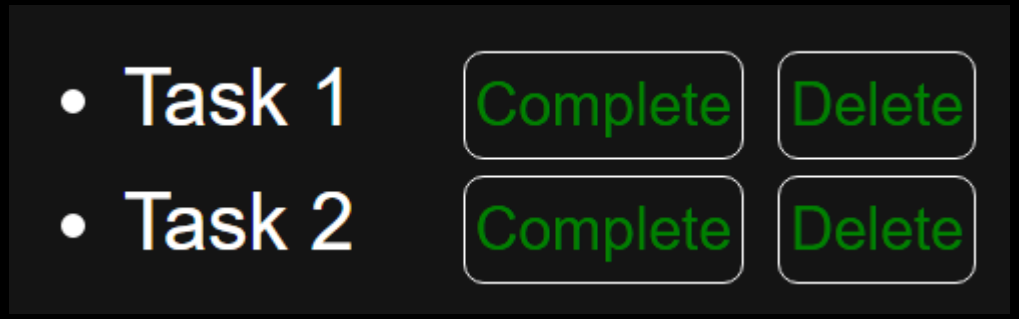
```
.actions li {
  color: green;
}


.tasks .task-items li {
  color: red;
}
```



Match on child elements – bad solution

```html
<div class="tasks">
  <ul class="task-items">
    <li>
      Task 1
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
    <li>
      Task 2
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
  </ul>
</div>
```

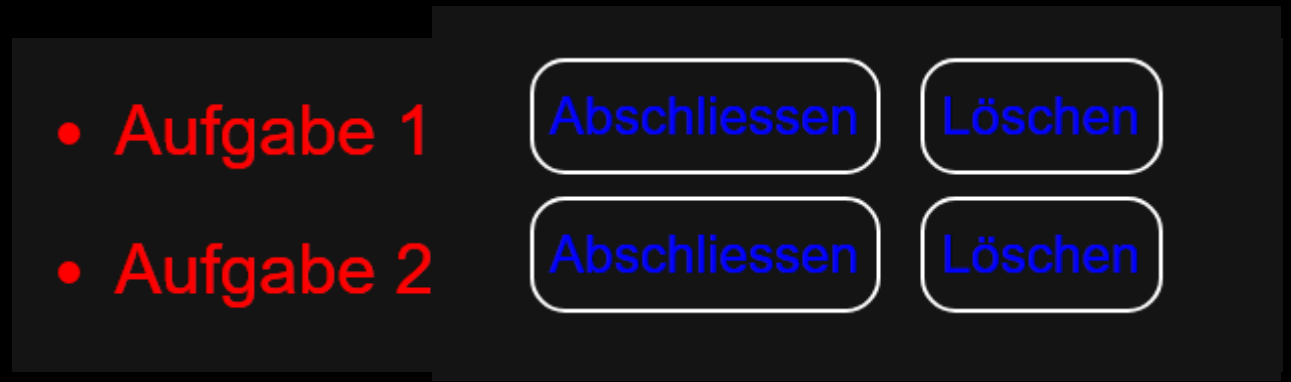
```css
.actions li {
  color: green !important;
}

.tasks .task-items li {
  color: red;
}
```



Match on child elements - better solution

```
<div class="tasks">
  <ul class="task-items">
    <li>
      Task 1
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
    <li>
      Task 2
      <ul class="actions">
        <li>Complete</li>
        <li>Delete</li>
      </ul>
    </li>
  </ul>
</div>
```

```
.actions li {
  color: blue;
}


.tasks .task-items > li {
  color: red;
}
```



Global CSS Scope

ISSUES ROUNDUP

BEM

NAMING CONVENTION (AND MORE)

Block, element, modifier

```
.title {}

.title__subline {}

.title__author {}

.title--xlarge {}

.title__subline--is-hidden {}
```

Block, element, modifier

```
.title {}

.title__subline {}

.title__author {}

.title--xlarge {}

.title__subline--is-hidden {}
```

Block, element, modifier

```
.title {}

.title__subline {}

.title__author {}

.title--xlarge {}

.title__subline--is-hidden {}
```

HTML

```
<div class="title title--xlarge">
  Modularize CSS
  <div class="title__subline">
      … different approaches
  </div>
  <div class="title__author">
      by Alex Silberschneider
  </div>
</div>
```

HTML

```
<div class="title title--xlarge">
  Modularize CSS
  <div class="title__subline">
    … different approaches
  </div>
  <div class="title__author">
    <div class="avatar">
        <div class="avatar__thumb"></div>
    </div>
    by Alex Silberschneider
  </div>
</div>
```

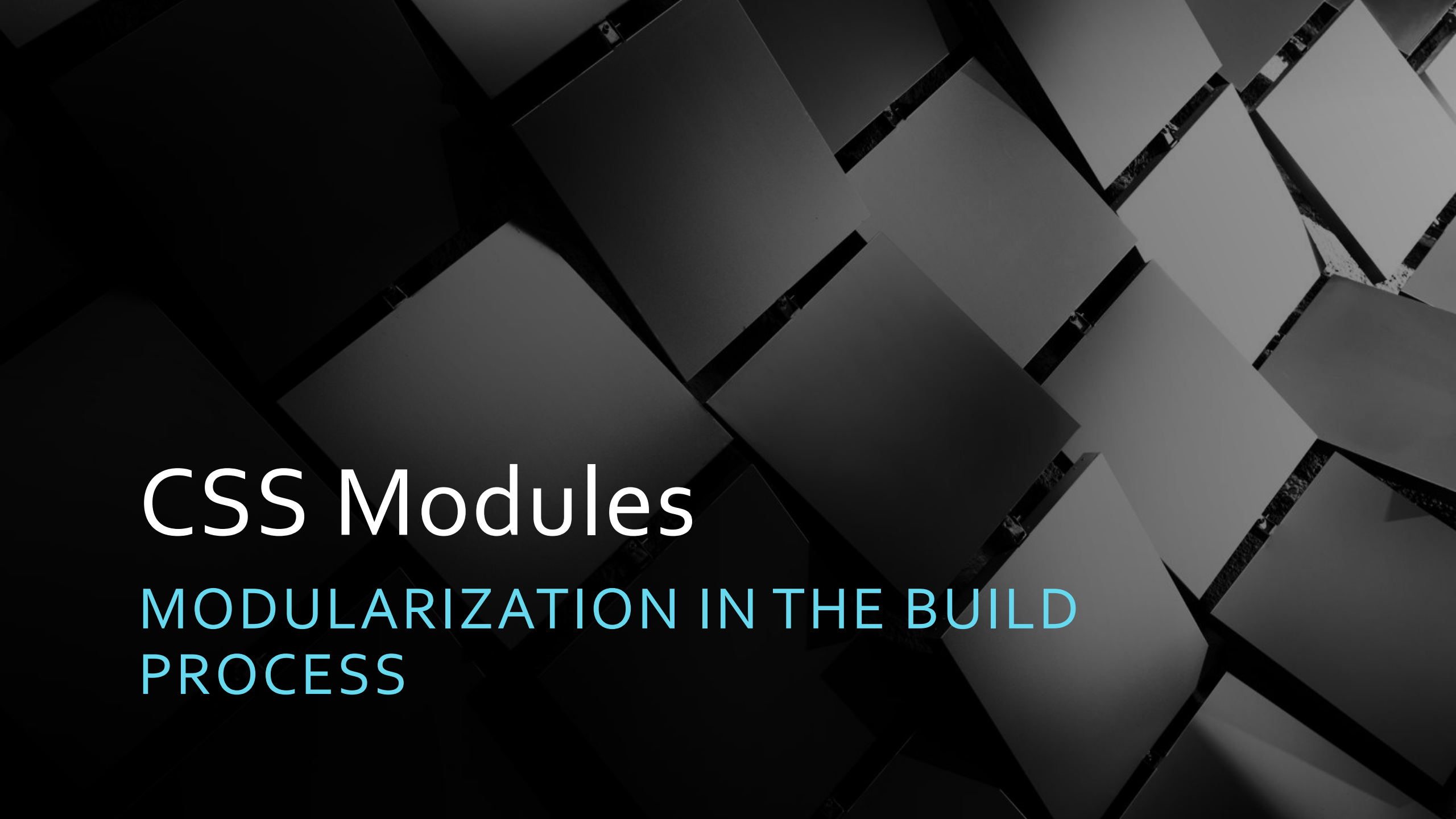
No nesting

```
<div class="title__subline">
    Subline text
    <div class="title__subline__date">
      01.12.2017
    </div>
 </div>
```

Example where Nesting is „OK“

```
.grid--is-editing .grid__cell {
  border: 1px solid black;
}
```

BEM benefits



CSS Modules

MODULARIZATION IN THE BUILD PROCESS

Initial code

```
<h1 class="bigtitle">
  A heading
</h1>

.bigtitle {
  color: red;
  font-size: 36px;
}
```

Instead of HTML, Javascript + Template string

```
import styles from "./styles.css";

element.innerHTML =
  `<h1 class="${styles.bigtitle}">
     A heading
   </h1>`;
```

Resulting output

```
<h1 class="_styles__bigtitle_4711156789">
  A heading
</h1>

._styles__bigtitle_4711156789 {
  color: red;
  font-size: 3rem;
}
```

React

```
import React from 'react'
import styles from './styles.css'

export default class CoolTitle extends React.Component {
  render() {
    return (
       <h1 className={styles.bigtitle}>{this.props.text}</h1>
    )
  }
}
```

Angular

```
import {Component} from '@angular/core';
import styles from './styles.css';

@Component({
    'selector': 'my-component',
    'template': `<h1 class="${styles.bigtitle}">
            Big title
        </h1>`
})
```

CSS Modules benefits

CSS in JS

GOODBYE CSS! (NOT REALLY)

Aphrodite

Aphrodite JS

```
import {StyleSheet, css} from
'aphrodite'


const styles = StyleSheet.create({
  bigtitle: {
    color: 'red',
    fontSize: '3rem',
    backgroundColor: `transparent',
    height: '2em',
    ':hover': {
      color: 'white',
      backgroundColor: `palevioletred'
    }
  }
});
```

```
const app =
document.getElementById('app');

app.innerHTML = `

  <h1
class="${css(styles.bigtitle)}">

    A heading again!

  </h1>`;


console.log(css(styles.bigtitle));

// "bigtitle_195i0wx"
```

Aphrodite JS

```
import {StyleSheet, css} from
'aphrodite'


const styles = StyleSheet.create({
  bigtitle: {
    color: 'red',
    fontSize: '3rem',
    backgroundColor: 'transparent',
    height: '2em',
    ':hover': {
      color: 'white',
      backgroundColor: 'palevioletred'
    }
  }
});
```

```
console.log(css(styles.bigtitle));

// "bigtitle_195i0wx"


const app =
document.getElementById('app');

app.innerHTML = `

  <h1
class="${css(styles.bigtitle)}">

    A heading again!

  </h1>`;
```

A heading again!

Aphrodite CSS

```
<style type="text/css" data-aphrodite="">
.bigtitle_wnu5ld {
    background-color:red !important;
    color:blue !important;
    height:2em !important;
}
.bigtitle_wnu5ld:hover {
    color:white !important;
    background-color:black !important;
}
</style>
```

CSS in JS roundup


Shadow DOM
A FUTURE POSSIBILITY

Simple example

HTML

```
<h1>Default Title</h1>
<div class="my-host"></div>
```

CSS

```
h1 {
  color: blue !important;
}
```

```
const myElement =
document.querySelector('.my-host');

const shadow =
myElement.attachShadow({
  mode: 'closed'
});
shadow.innerHTML = `
  <style>
    h1 {
      color: red;
    }
  </style>
  <h1>Title in Shadow DOM</h1>
`;
```

How to apply styles

HTML

```
<h1>Default Title</h1>
<div class="my-host"></div>
```

CSS

```
h1 {
  color: blue !important;
}

html {
  --my-custom-h1-color: green;
}
```

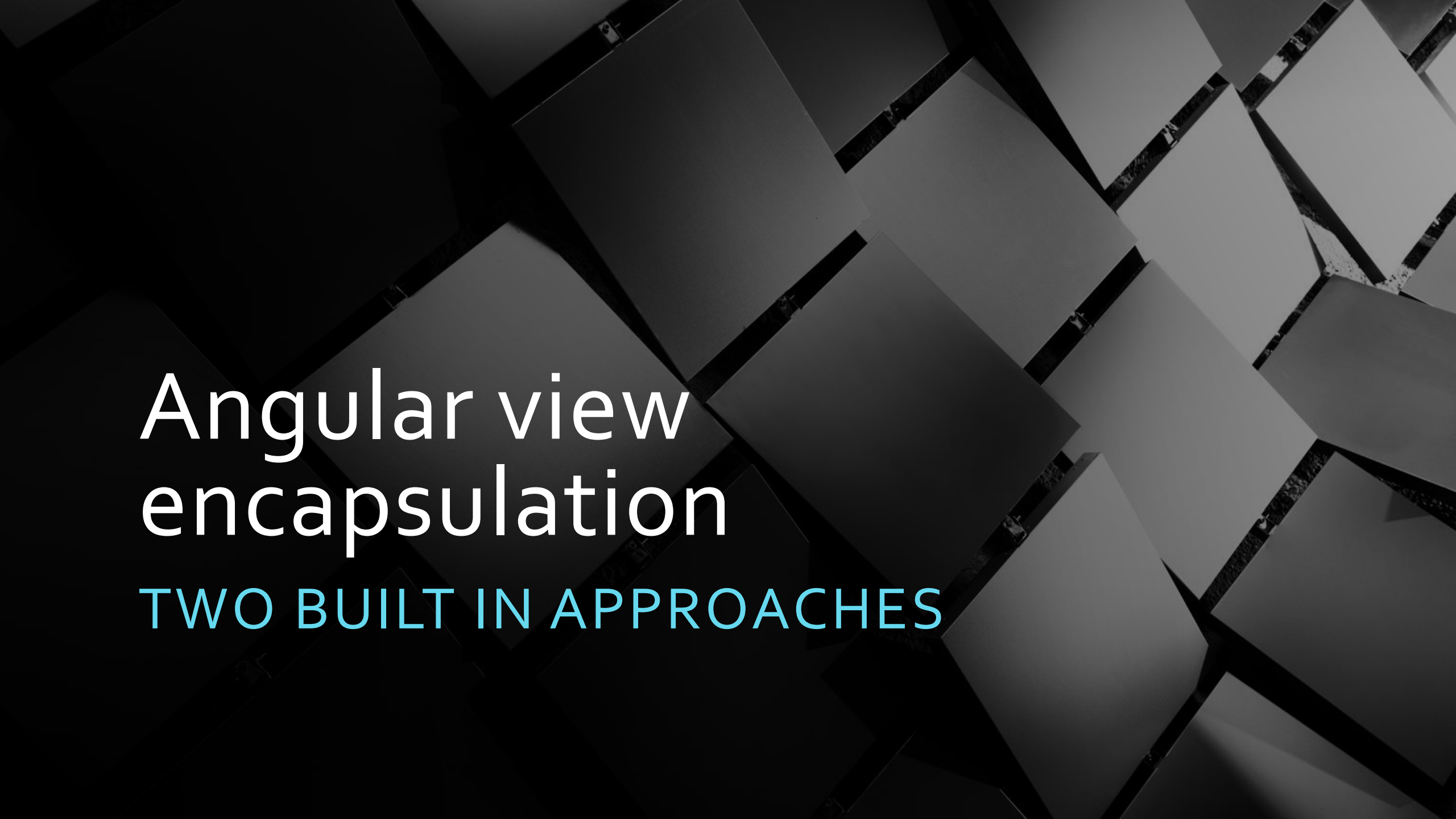
```
let myElement =
document.querySelector('.my-host');

let shadow =
myElement.attachShadow({
  mode: 'closed'
});
shadow.innerHTML = `
  <style>
    h1 {
 color: var(--my-custom-h1-color,
red);
    }
  </style>
  <h1>Title in Shadow DOM</h1>
`;
```

Default Title

Title in Shadow DOM

Shadow DOM roundup

SUPPORT 56%
HTTP://CANIUSE.COM/#FEAT=SHADOWDOMV1


Angular view encapsulation
TWO BUILT IN APPROACHES

„Emulated“ - default

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

@Component({

selector: 'app-first-component',

templateUrl: './first-
component.component.html',

styleUrls: ['./first-
component.component.css'],

encapsulation:
ViewEncapsulation.Emulated

})

export class
FirstComponentComponent {}
```

```
h1 {
    color: red;
}
```

First component H1

Another components H1

„Emulated“ – how does this work

```
<h1 _ngcontent-
c1="">
First component H1
</h1>

h1[_ngcontent-c1] {
color: red;
}
```

First component H1

Another components H1

„Native“ – Shadow DOM

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

@Component({

selector: 'app-first-component',

templateUrl: './first-
component.component.html',

styleUrls: ['./first-
component.component.css'],

encapsulation:
ViewEncapsulation.Native

})

export class
FirstComponentComponent {}
```

First component H1

Another components H1

„Native“ – how does this work

```
import { Component,
ViewEncapsulation

 } from '@angular/core';


@Component({

selector: 'app-root',

templateUrl:
'./app.component.html',

styleUrls:
['./app.component.css'],

encapsulation:
ViewEncapsulation.Native

})

export class AppComponent {}
```

```
▼<app-first-component> == $0
  ▼#shadow-root (open)
      <style>h1 {
          color: red;
      }</style>
      <style></style>
      <h1>
          First component H1
      </h1>
  </app-first-component>
```

First component H1

Another components H1

Many possibilities

EVERY METHOD HAS ITS PROS AND CONS

Thank you for listening!

Appendix

Link CSS in general

- https://philipwalton.com/articles/side-effects-in-css/
- http://thesassway.com/intermediate/avoid-nested-selectors-for-more-modular-css
- http://www.standardista.com/css3/css-specificity/
- http://thesassway.com/beginner/the-inception-rule
- https://csswizardry.com/2012/11/code-smells-in-css/

Links BEM

- https://en.bem.info/
- https://css-tricks.com/bem-101/
- https://csswizardry.com/2013/01/mindbemding-getting-your-head-round-bem-syntax/
- https://www.smashingmagazine.com/2016/06/battling-bem-extended-edition-common-problems-and-how-to-avoid-them/
- https://www.sitepoint.com/bem-smacss-advice-from-developers/
- https://medium.com/@stowball/bemantic-dry-like-you-mean-it-133ea3843d98
- https://www.smashingmagazine.com/2014/07/bem-methodology-for-small-projects/
- https://mattstauffer.co/blog/organizing-css-oocss-smacss-and-bem/
- https://csswizardry.com/2015/03/more-transparent-ui-code-with-namespaces/

Links CSS Modules

- https://css-tricks.com/css-modules-part-1-need/ +2+3 (Attention: Breaking webpack changes in the tutorial! Ask me..)
- https://glenmaddern.com/articles/css-modules
- https://www.sitepoint.com/understanding-css-modules-methodology/
- https://github.com/nkbt/ng-modular
- https://vuejs.org/v2/guide/single-file-components.html
- https://github.com/joaogarin/css-modules-angular2

Links CSS in JS

- https://github.com/MicheleBertoli/css-in-js
- https://medium.freecodecamp.org/css-in-javascript-the-future-of-component-based-styling-70b161a79a32
- http://engineering.khanacademy.org/posts/aphrodite-inline-css.htm
- https://github.com/airbnb/javascript/tree/master/css-in-javascript
- https://www.styled-components.com/docs
- https://github.com/threepointone/glamor
- https://medium.com/@gajus/stop-using-css-in-javascript-for-web-development-fa32fb873dcc ;)

Links Shadow DOM

- http://caniuse.com/#feat=shadowdomv1
- https://css-tricks.com/playing-shadow-dom/
- https://frontend.namics.com/2014/05/27/web-components-shadow-dom/
- https://developers.google.com/web/fundamentals/architecture/building-components/shadowdom
- https://www.smashingmagazine.com/2016/12/styling-web-components-using-a-shared-style-sheet/
- https://www.polymer-project.org/2.0/start/first-element/step-5

Pens

- https://codepen.io/silberxander/