How Styles Work in Angular

Brian Treese
CHIEF OF USER EXPERIENCE AT SOCREATE.IT

@brianmtreese

HTML

JavaScript

CSS


00:00

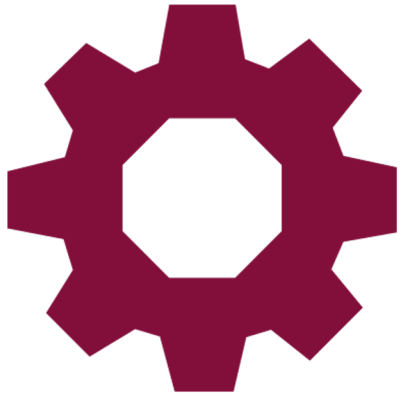
Encapsulation & Isolation

Bundle for Reuse

Use in Multiple Applications

Web Components

Custom Elements



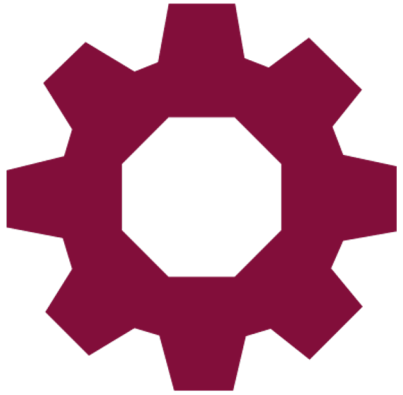
HTML Templates

Shadow DOM

```
<my-custom-element>
    Put content and markup here
</my-custom-element>
```

Custom Elements

```
<template id="template">
    Put content and markup here
</template>
```

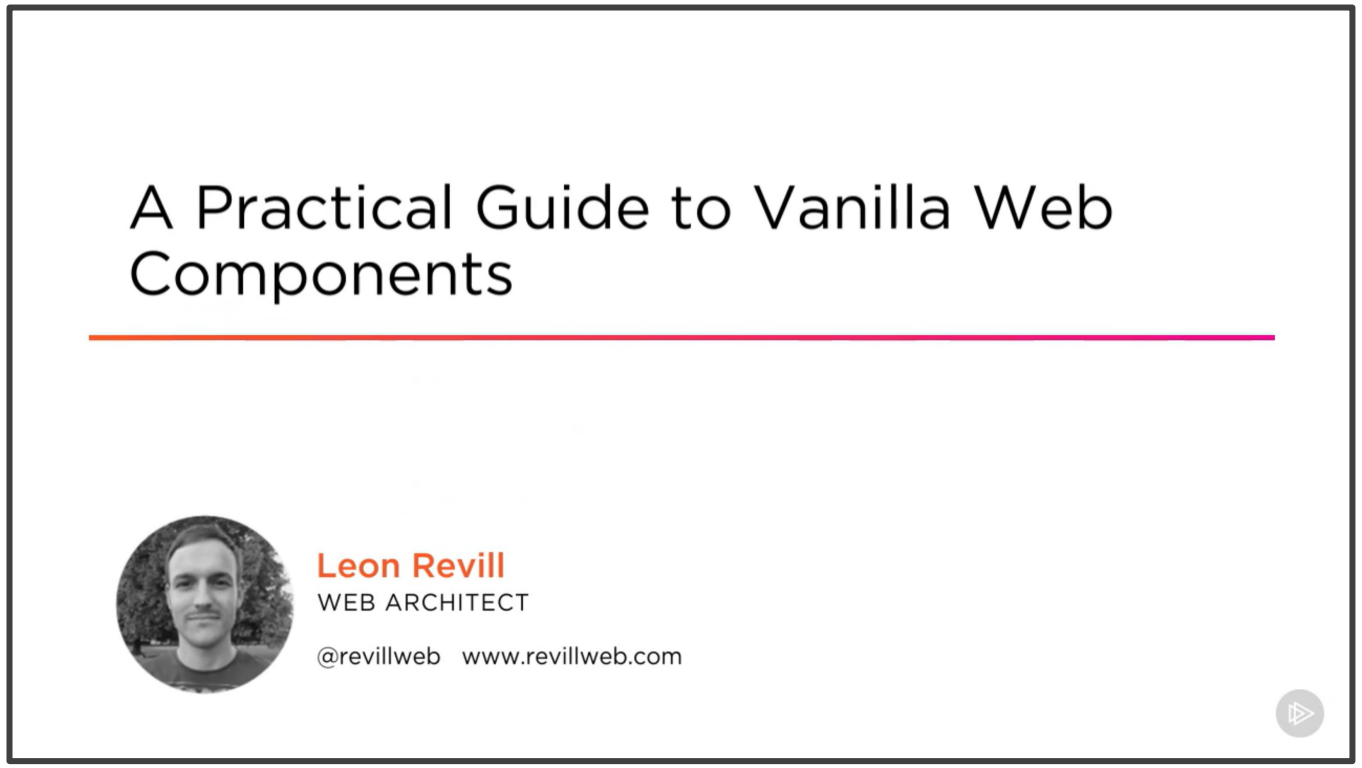


HTML Templates

```
<html>
    <my-custom-element>
        #shadow-root (user-agent)
            Content and markup ends up here
    </my-custom-element>
```

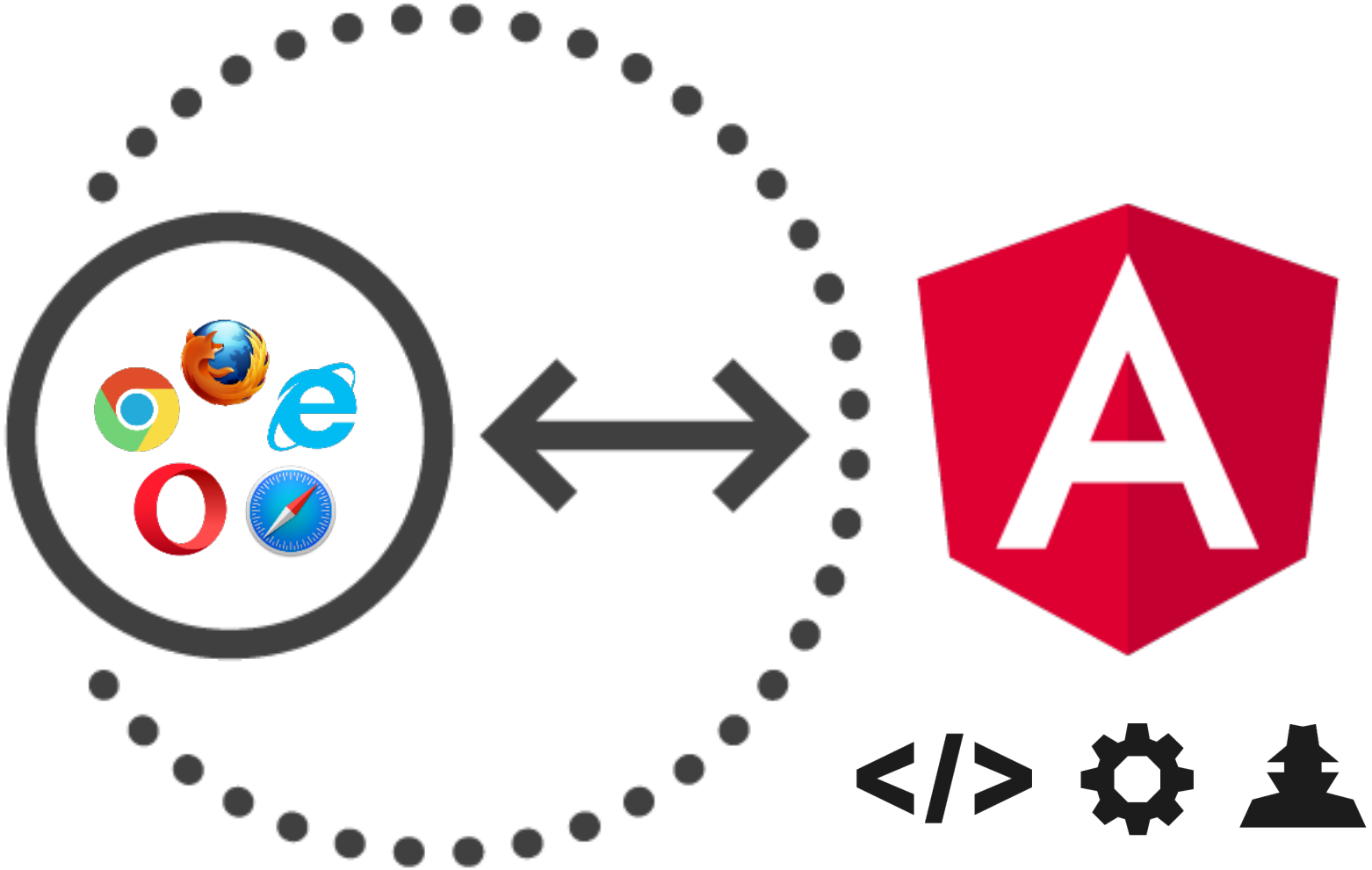
Shadow DOM

Want More on Web Components?



https://app.pluralsight.com/library/courses/vanilla-web-components-practical-guide/table-of-contents

Angular Fills Some Gaps



Coming Up...

Basic Component Structure

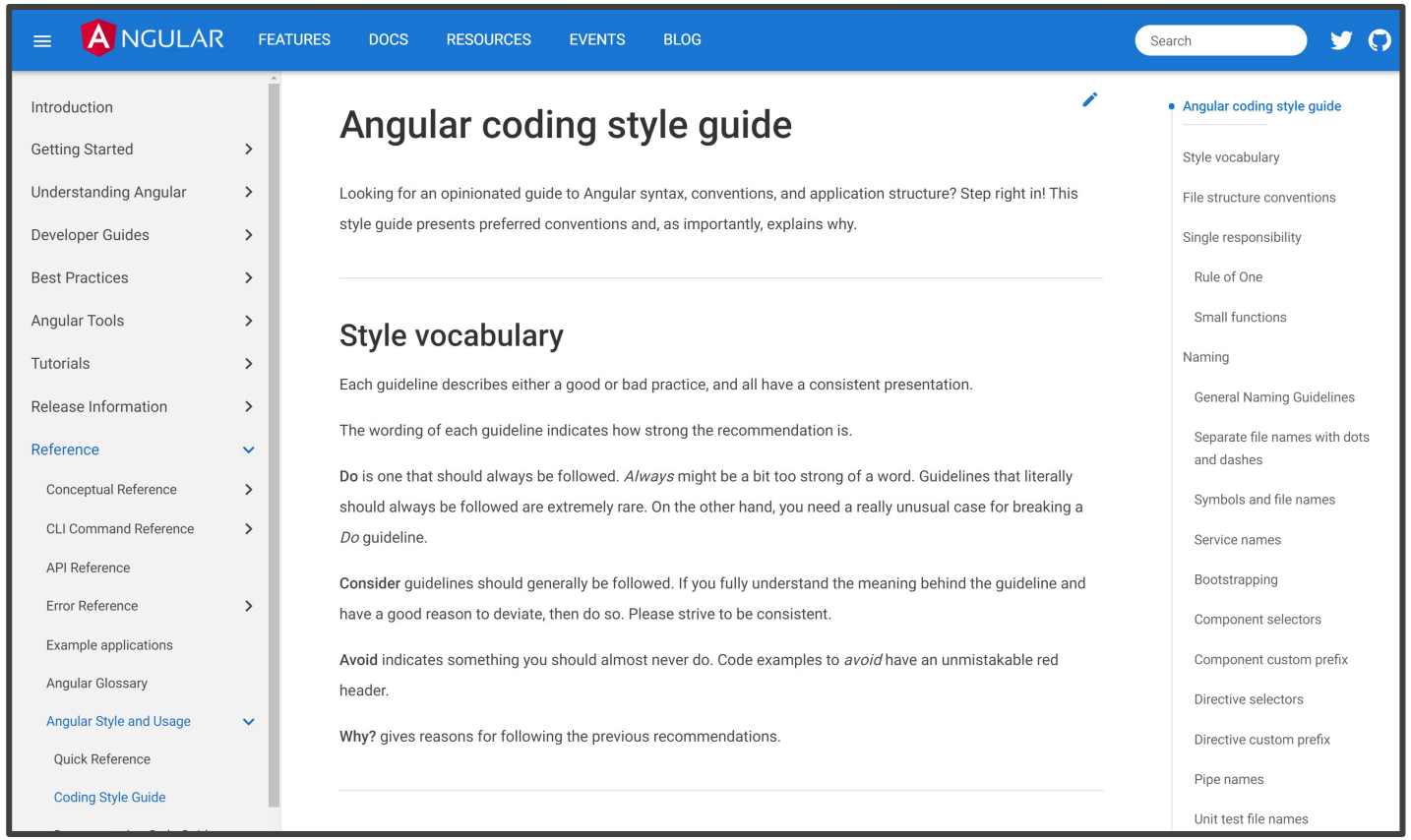
View Encapsulation

Adding Styles

Scoping CSS Selector Emulation

Angular Component Structure

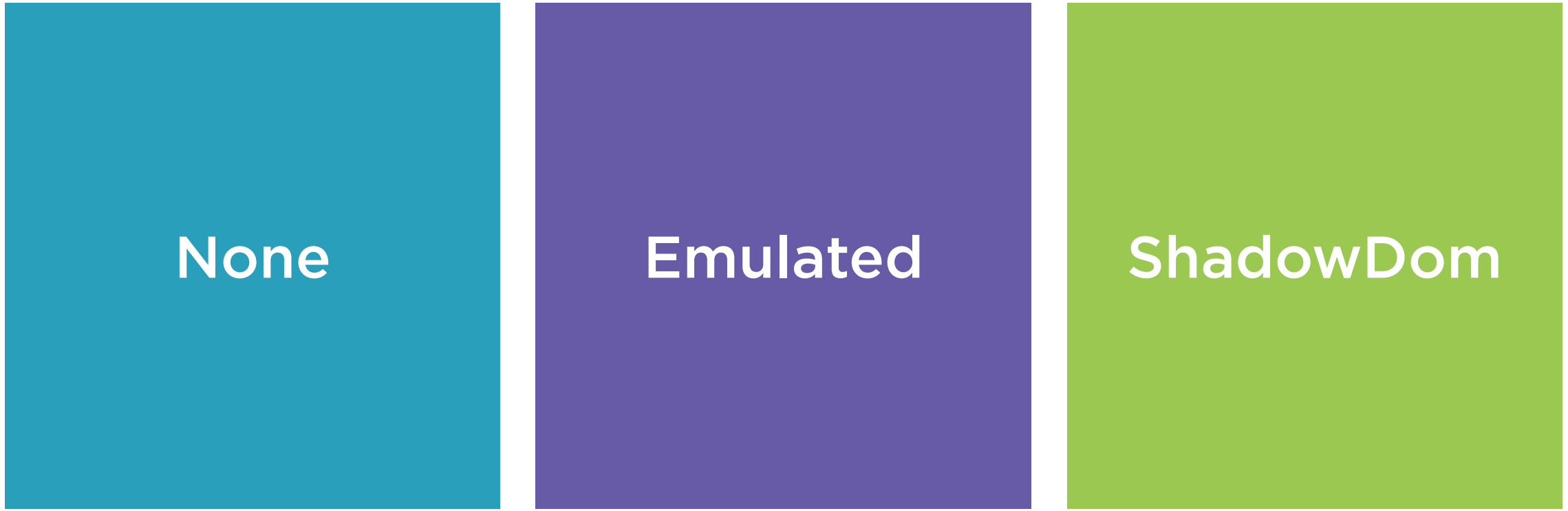
Angular Style Guide



https://angular.io/guide/styleguide

View Encapsulation

View Encapsulation Modes



In a Traditional Web Application...

```
<head>
    <link rel="stylesheet" href="/path-to/stylesheet.css">
</head>
```

Don’t Add Styles Tied to These Attributes

View Encapsulation Modes: None & ShadowDom

This Feels Wrong

ShadowDom

Native Shadow DOM

Emulated Mode

Component Styles

Other Ways to Add Styles

Embedded

<link>

@import

Inline

styleUrls

Styles In the Template

Linking From an External File

CSS Imports

Inline Styles

styleUrls

~~<link rel="stylesheet" href="/path-to/stylesheet.css">~~

```
styleUrls:['stylesheet.css']
```

Syntax Highlighting & Code Completion

Syntax Highlighting & Code Completion

The Same but Different

styleUrls

```
styleUrls: ['example.component.scss']
```

link

```
<link rel="stylesheet" href="/path/stylesheet.css">
```

templateUrl

CSS Preprocessors

Emulated CSS Selectors

CSS Scoping

:host

:host-context

::ng-deep

“Host”

```
<saa-app-nav>
    <ul>...</ul>
</saa-app-nav>
```

“Shadow Tree”

```
<saa-app-nav>
    <ul>...</ul>
</saa-app-nav>
```

:host

```
:host {
    background: #2A9FBC;
    border-radius: 0.5em;
    margin: 1.5em 0;
    padding: 0;
}
```

:host-context

```
<div class="container">
    <saa-app-nav>
        <ul>...</ul>
    </saa-app-nav>
</div>
```

:host-context

```
:host-context(.container) {
    background: #2A9FBC;
    border-radius: 0.5em;
    margin: 1.5em 0;
    padding: 0;
}
```

Why Use :host?

Why Not :host-context?

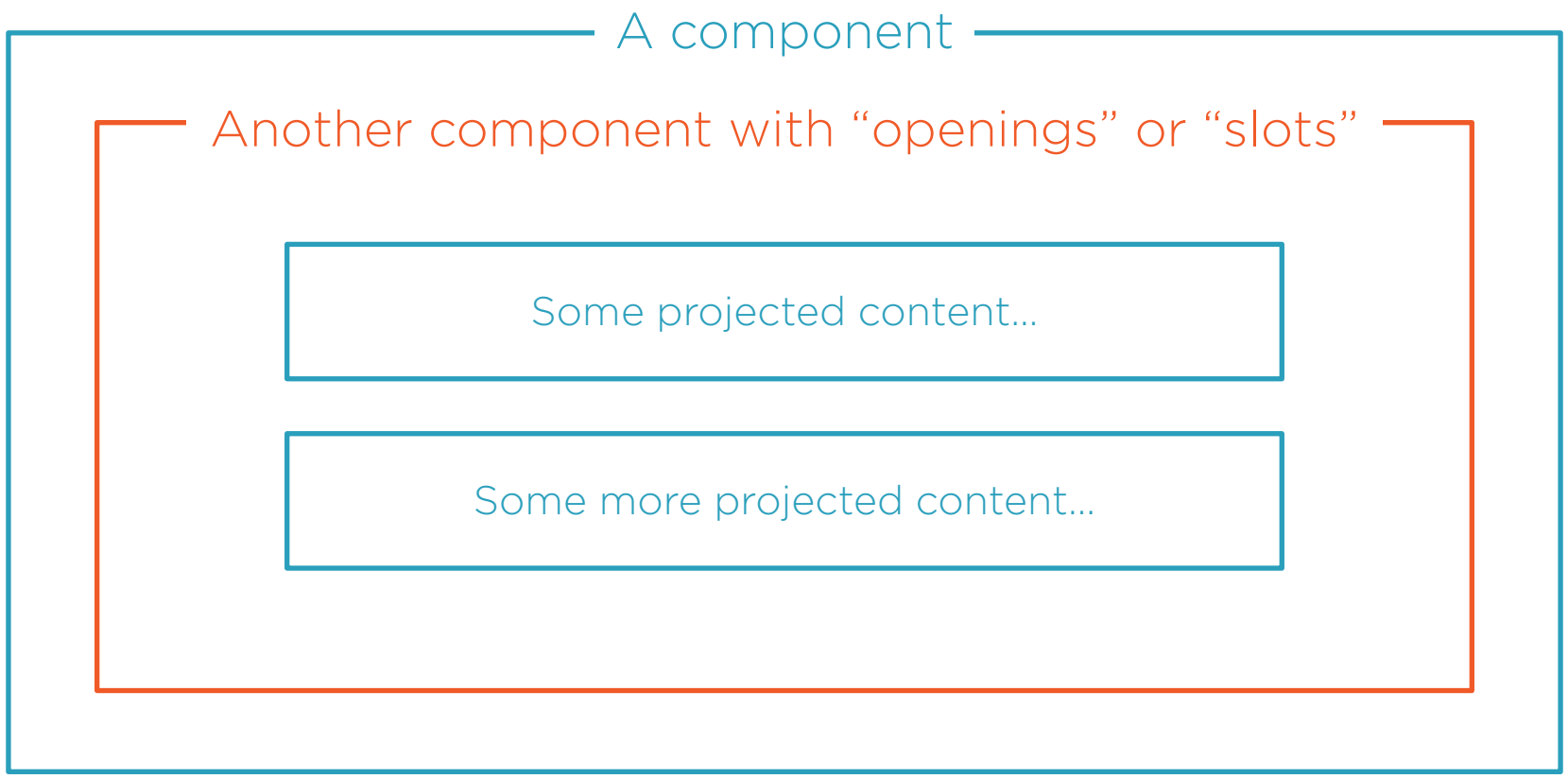
We Don’t Always Know!

:host-context

Use it Purposefully!


::ng-deep

Content Projection



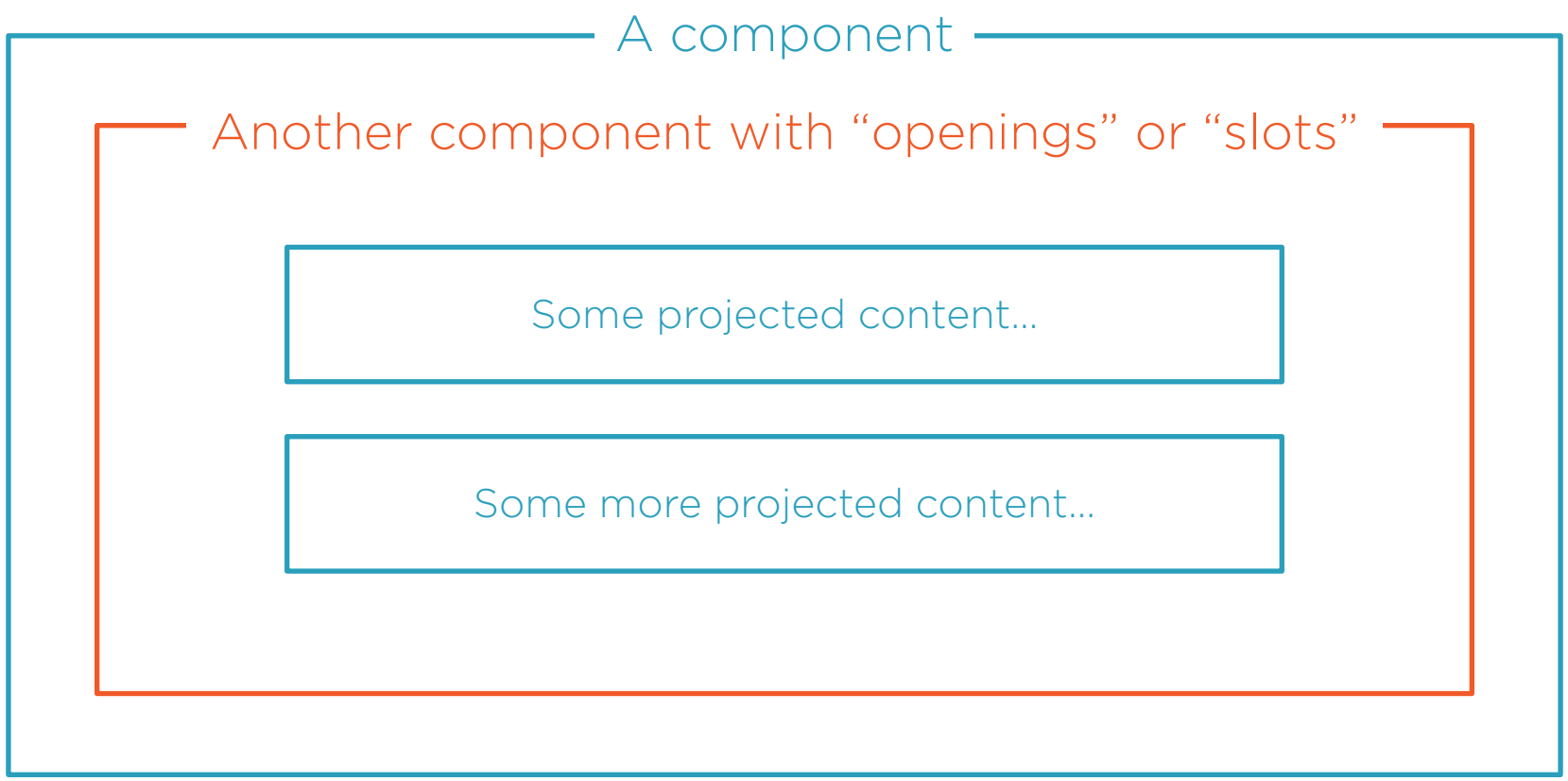
Content Projection

```
<div>
    This is where content goes
</div>
```

Content Projection

```
<my-component>
   This is also where content goes
</my-component>
```

Content Belongs to Parent



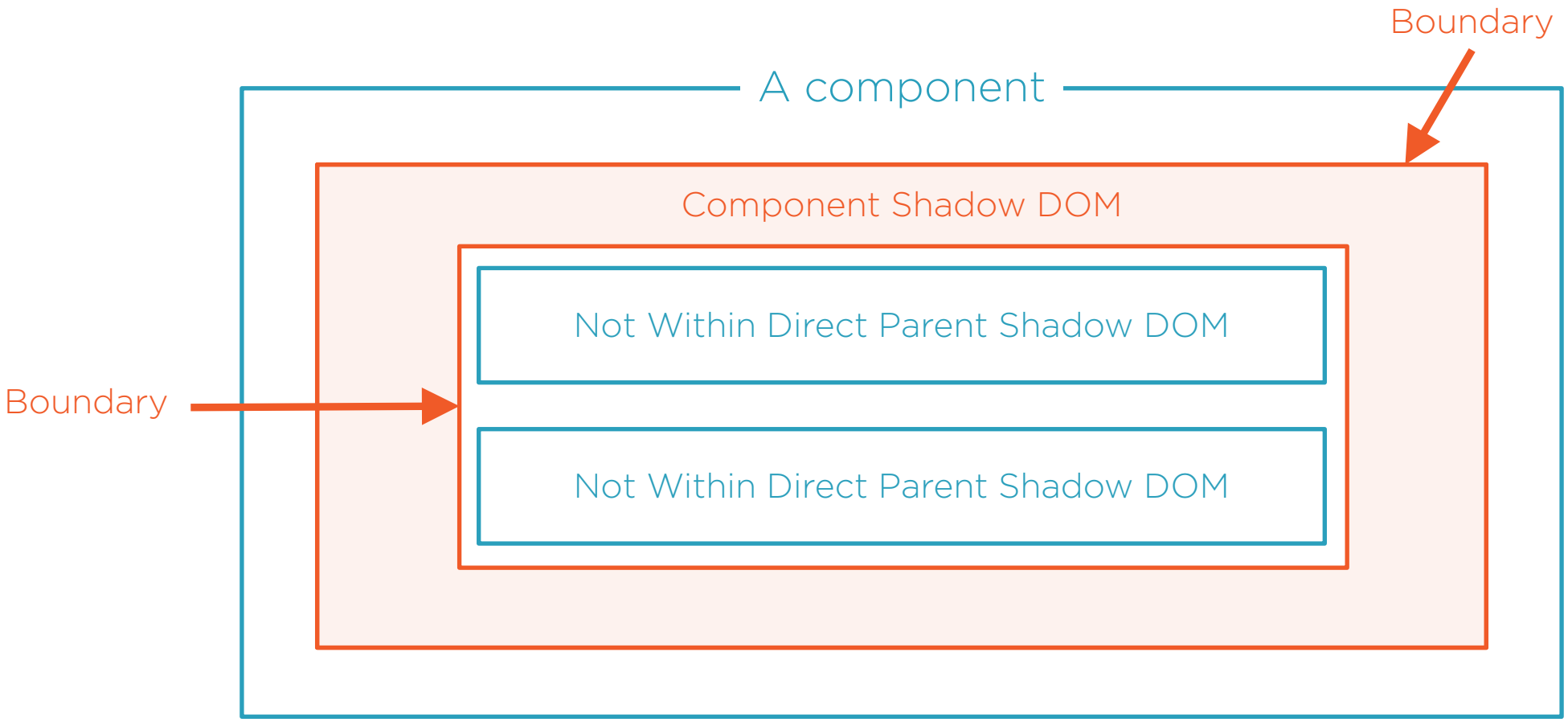
Pop Up Dialog

We Don’t Know

Pop Up Dialog

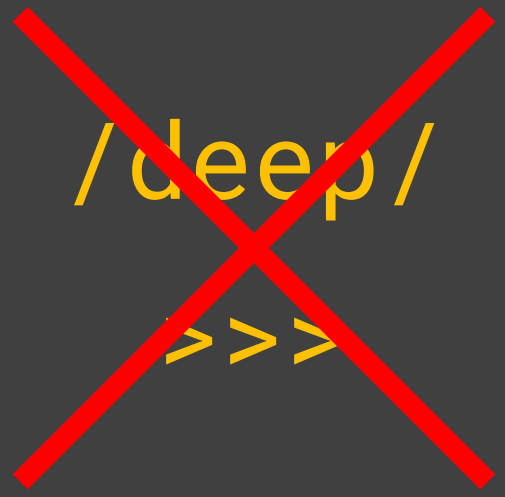
We Do Know

Shadow Boundary




Yikes!

CSS Scoping Module Spec



Deprecated

Where We've Been...

Styling Angular Components

Shadow DOM & CSS Scoping Emulation

Where We're Headed...

Wrap Up How Styles Work

Create a Scalable & Maintainable System

Summary

Style Encapsulation

Angular Shadow DOM/CSS Scoping Module Emulation

Custom Elements

Component HTML Templates

View Encapsulation: Emulated, ShadowDom, & None

Adding Styles: Styles Property, Embedded, Linked, Imported, Inline, External

Using Preprocessors

:host, :host-context, ::ng-deep

We Need to Rethink...

How we craft & organize styles

How we style for predictability, scale, & maintainability