Manipulating the UI

David Mann

@MannD | www.HeirloomSoftware.com

Topics

View Encapsulation

Pipes

Directives

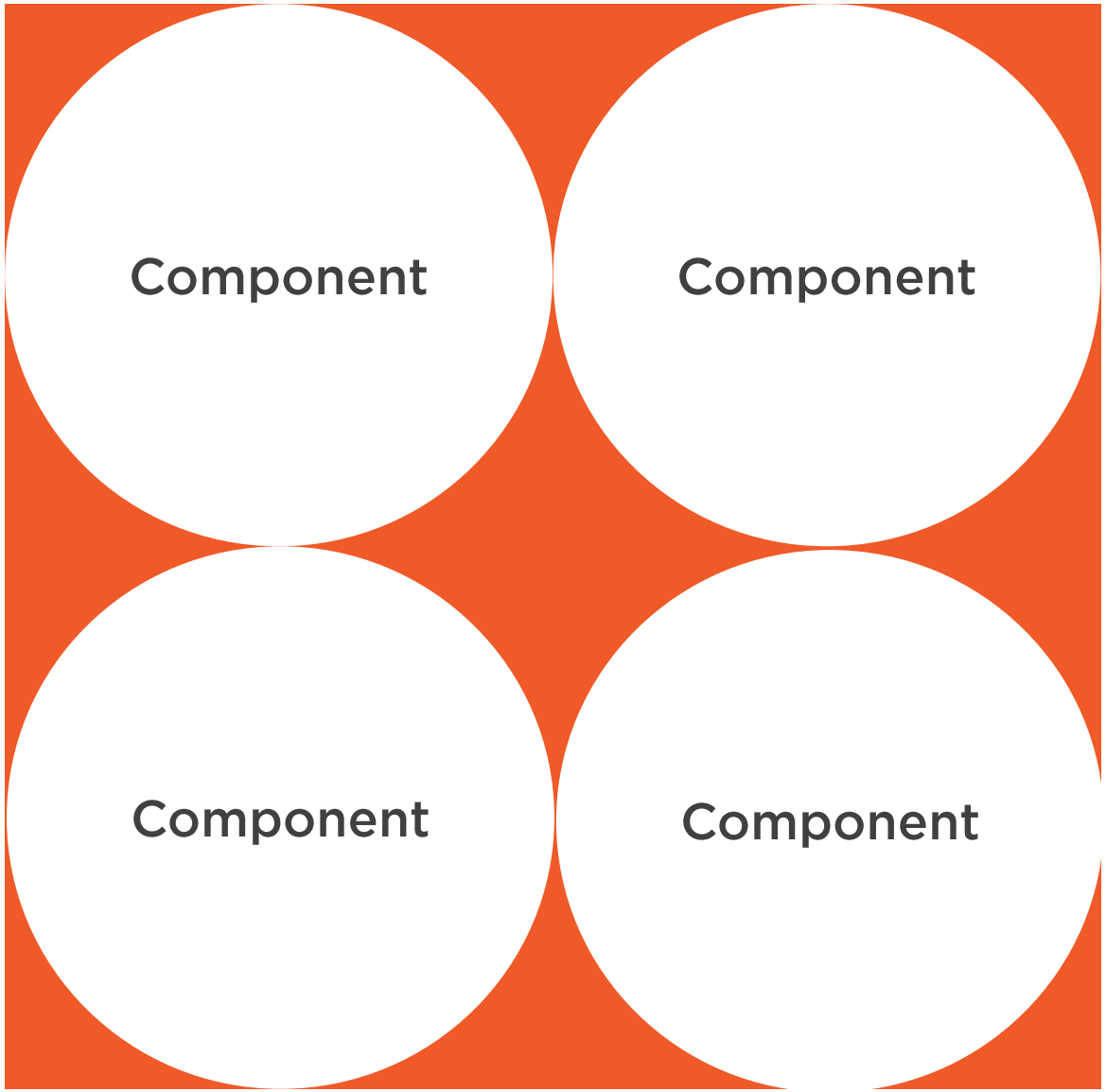
- HostBinding
- HostListener

nativeElement

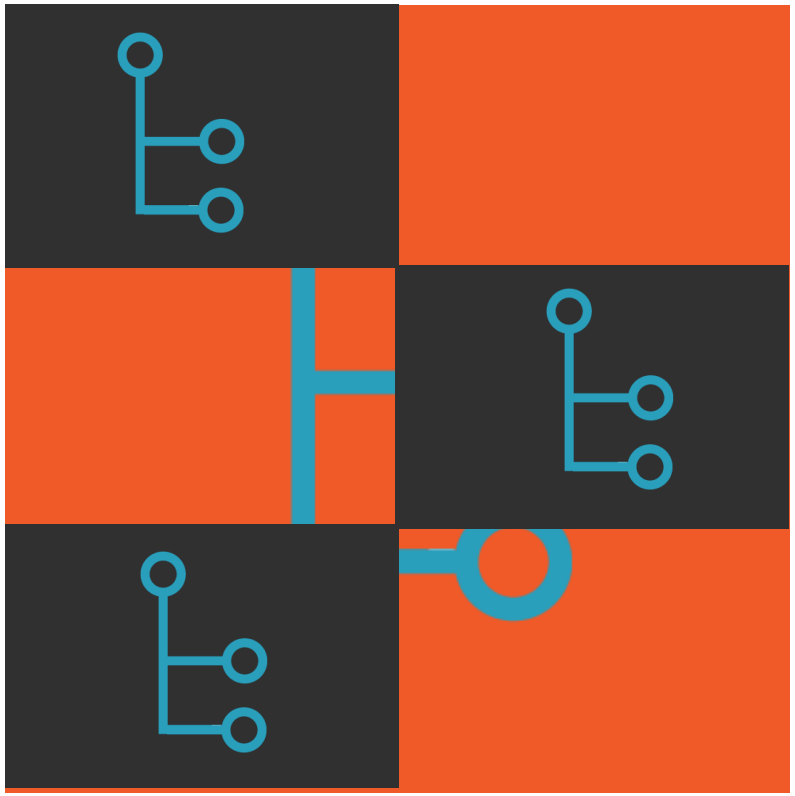
Renderer2

jQuery as an Anti-Pattern

View Encapsulation



Shadow DOM 101



DOM Isolation

View Encapsulation Options

Emulated

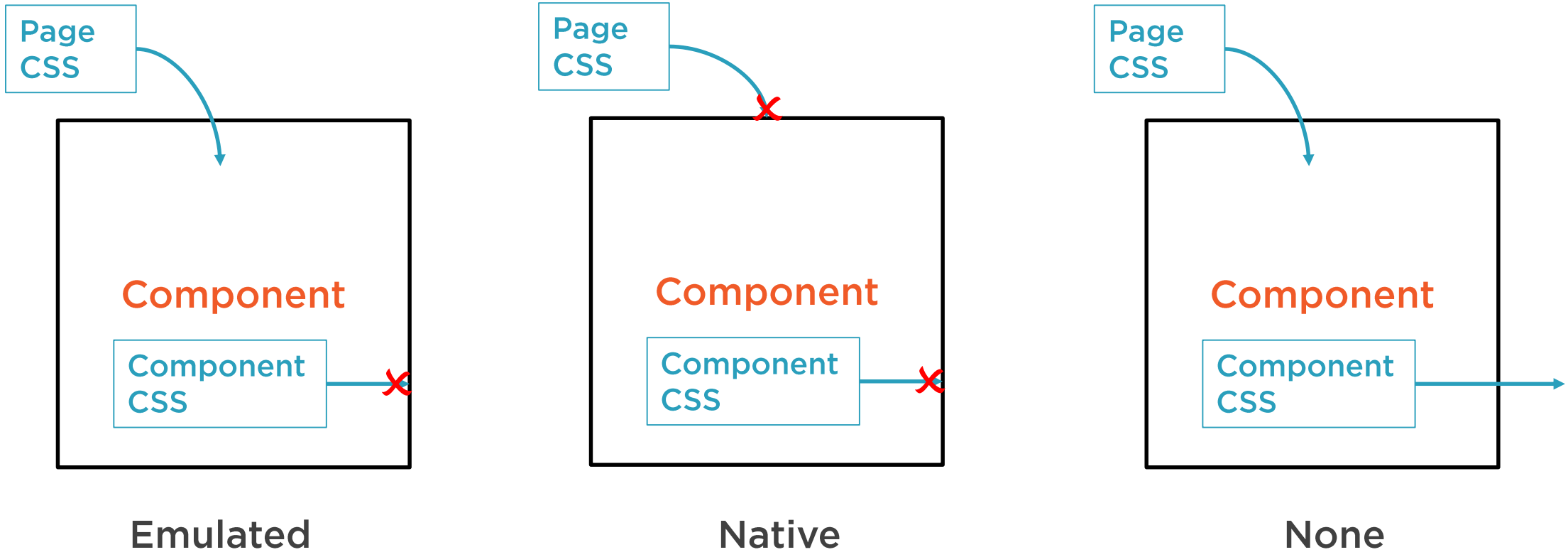
```
<span _ngcontent-c1 class="viewType">View</span>
```

```
.viewType[_ngcontent-c1] {
    text-decoration: ▶ underline;
}
```

Native

None

View Encapsulation Options



Emulated

Native

None

Shadow Piercing

::ng-deep

https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables

::ng-deep

```
::ng-deep.viewType {
    text-decoration: underline;
}
```

Pipes modify data for display (only)

Custom Pipe Use Case

LowerCase

UpperCase

TitleCase

Directive Selectors

```
@Directive({
 selector:'[hsPaywall]'
})
```

```
@Directive({
 selector: 'a'
})
```

```
@Directive({
 selector:'.class'
})
```

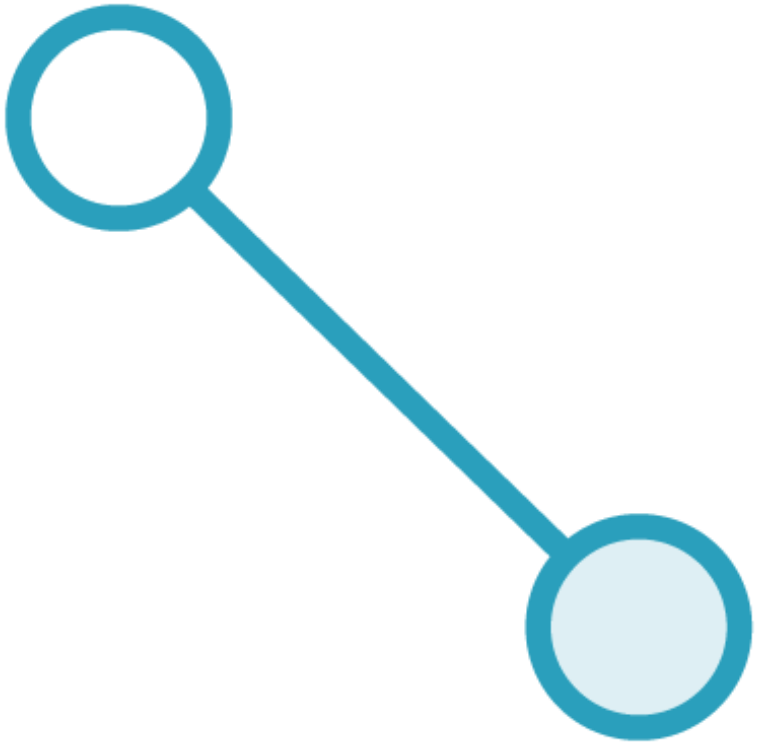
```
@Directive({
 selector:'.class:not(".class2")'
})
```

```
@Directive({
 selector:'.class, .class2'
})
```

```
@Directive({
 selector:'input[type="password"]'
})
```

```
@Directive({
 selector:'.class, [hsPaywall]'
})
```

Host Binding



Connect Directive property to parent (host)

- Class
- Style
- Attribute
- Property

Host

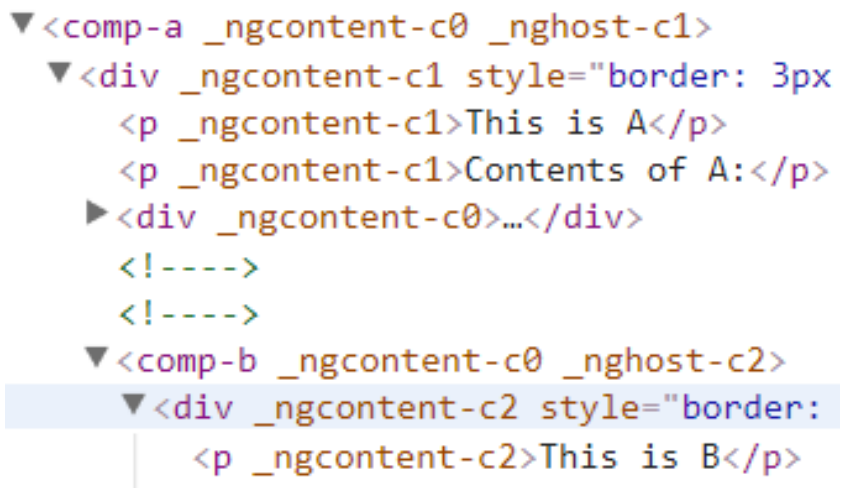
The element containing our Directive

```
<div myDirective>
```

Host Listener

Respond to Host Events

nativeElement



DOM Node (in browser clients)

Use It

Know the implications

USE WITH CAUTION

Use this API as the last resort when direct access to DOM is needed. Use templating and data-binding provided by Angular instead. Alternatively you can take a look at `Renderer2` which provides API that can safely be used even when direct access to native elements is not supported.

Relying on direct DOM access creates tight coupling between your application and rendering layers which will make it impossible to separate the two and deploy your application into a web worker.

Source: https://angular.io/api/core/ElementRef

Tight Coupling

~~Angular Universal~~

~~Web Worker~~

BUT...

~~Tight~~ Loose Coupling

Angular Universal ✓

Web Worker ✓

```
this.elmB.nativeElement.style.color = 'red’;

this.elmB.nativeElement.addEventListener('click’,
        () => this.doClick() );

this.elmB.nativeElement.setAttribute('myAttr', 'true’);

let sib = this.elmB.nativeElement.nextSibling;
```

Improper Use of nativeElement

Don’t do this!

Direct access to nativeElement members

Using nativeElement Safely

~~Angular Universal~~

~~Web Worker~~

BUT...

Using nativeElement Safely

~~Angular Universal~~

~~Web Worker~~

Renderer2

Renderer2

```
class Renderer2 {
  get data: {[key: string]: any}
  destroy(): void
  createElement(name: string, namespace?: string|null): any
  createComment(value: string): any
  createText(value: string): any
  destroyNode: ((node: any) => void)|null
  appendChild(parent: any, newChild: any): void
  insertBefore(parent: any, newChild: any, refChild: any): void
  removeChild(parent: any, oldChild: any): void
  selectRootElement(selectorOrNode: string|any): any
  parentNode(node: any): any
  nextSibling(node: any): any
  setAttribute(el: any, name: string, value: string, namespace?: string|null): void
  removeAttribute(el: any, name: string, namespace?: string|null): void
  addClass(el: any, name: string): void
  removeClass(el: any, name: string): void
  setStyle(el: any, style: string, value: any, flags?: RendererStyleFlags2): void
  removeStyle(el: any, style: string, flags?: RendererStyleFlags2): void
  setProperty(el: any, name: string, value: any): void
  setValue(node: any, value: string): void
  listen(target: 'window'|'document'|'body'|any, eventName: string, callback: (event: any) => boolean | void): () => void
}
```

https://angular.io/api/core/Renderer2

~~this.elmB.nativeElement.style.color = 'red';~~

~~this.elmB.nativeElement.addEventListener('click', () => this.doClick());~~

~~this.elmB.nativeElement.setAttribute('myAttr', 'true');~~

~~let sib = this.elmB.nativeElement.nextSibling;~~

```
this.ren2.setStyle(this.elmB.nativeElement, 'color', 'red');

this.ren2.listen(this.elmB.nativeElement, 'click', () => this.handleClick());

this.ren2.setAttribute(this.elmB.nativeElement, 'myAttr', 'true');

let sib = this.ren2.nextSibling(this.elmB.nativeElement);
```

Not fighting the framework

Better Angular code

jQuery?

What is jQuery?

DOM Manipulation ✓

Event Management ✓

AJAX ✓

Async (Promise) ✓

Animation ✓

Plug-Ins ?

jQuery is not bad

jQuery doesn’t belong in an Angular project

Make a choice

Key Takeaways

View Encapsulation

Pipe

Directive Selectors

Host Binding/Listening

nativeElement/Renderer2

~~jQuery~~

Next Up

Dynamic Components

Content Projection