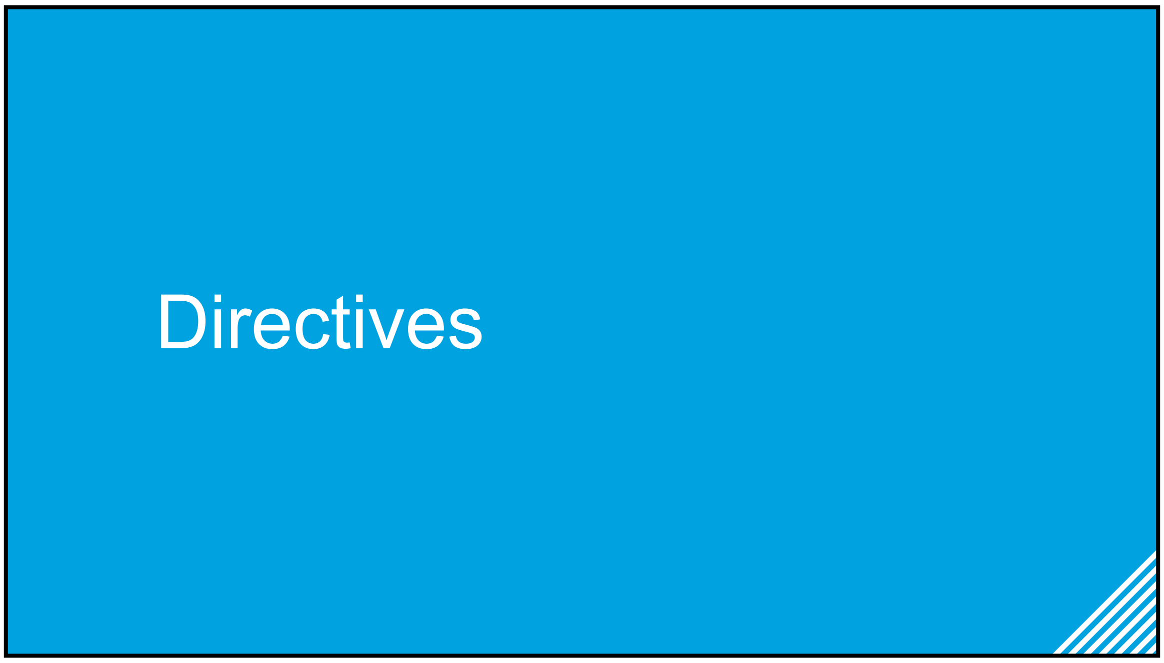
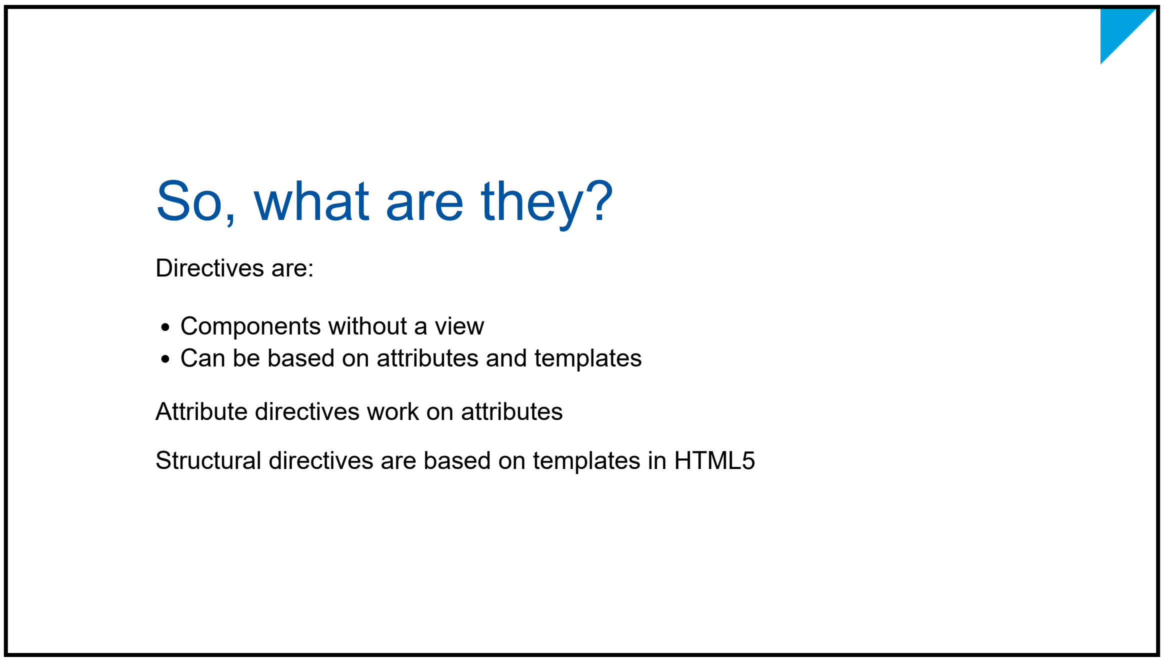


Directives



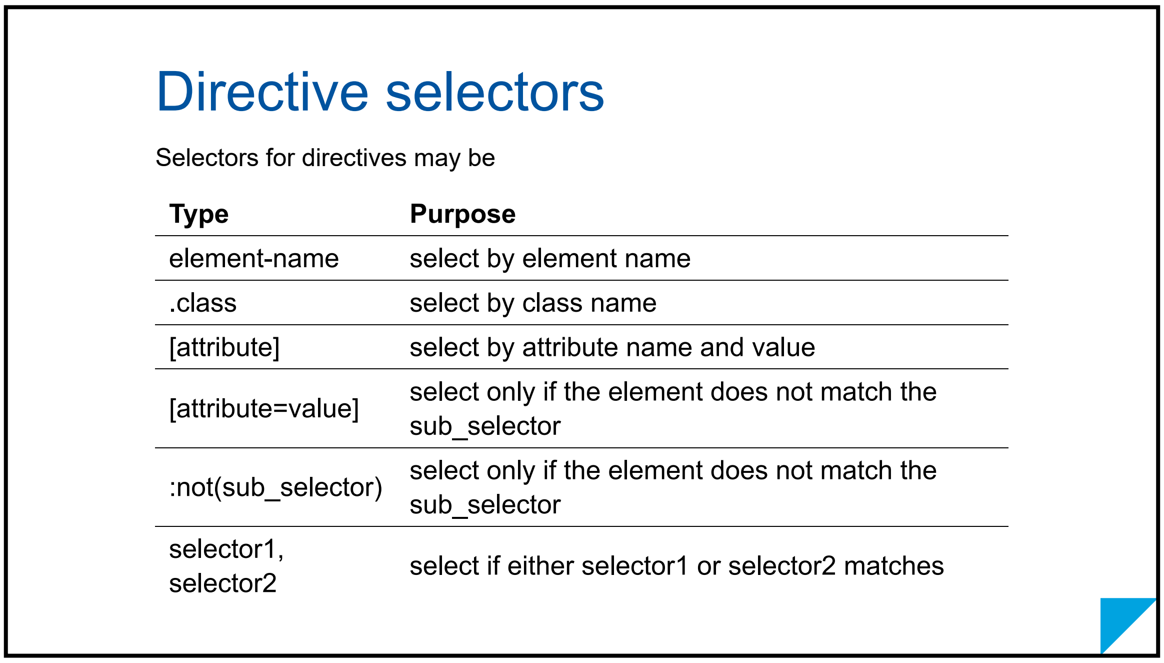
So, what are they?

Directives are:

- Components without a view
- Can be based on attributes and templates

Attribute directives work on attributes

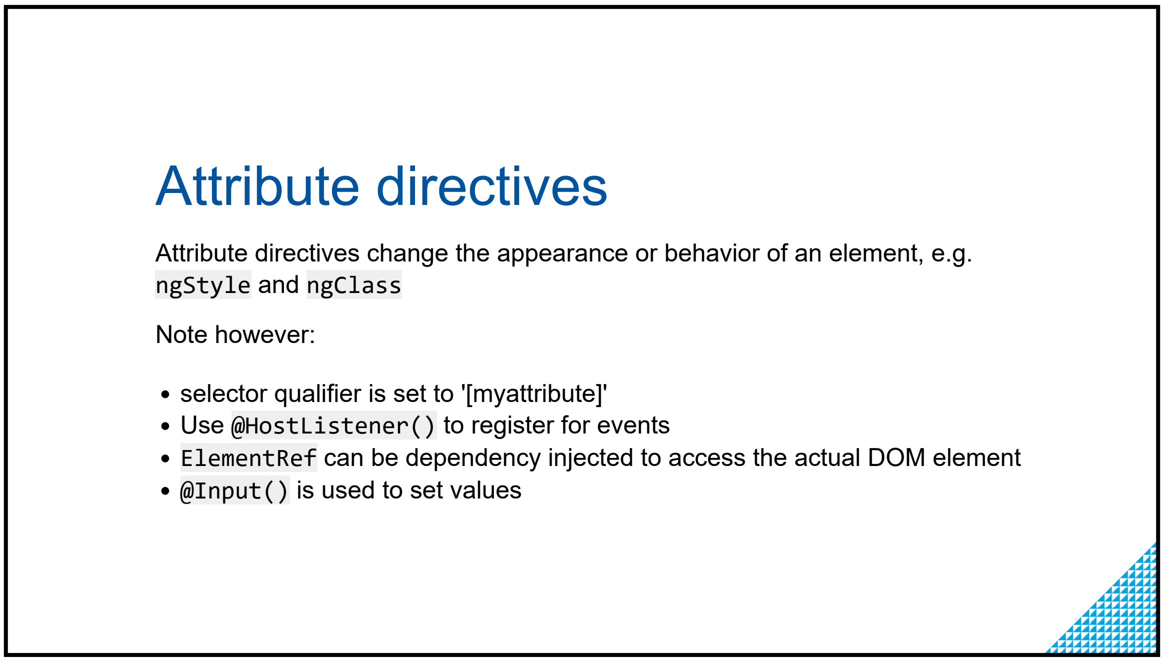
Structural directives are based on templates in HTML5



Directive selectors

Selectors for directives may be

Type	Purpose
element-name	select by element name
.class	select by class name
[attribute]	select by attribute name and value
[attribute=value]	select only if the element does not match the sub_selector
:not(sub_selector)	select only if the element does not match the sub_selector
selector1, selector2	select if either selector1 or selector2 matches

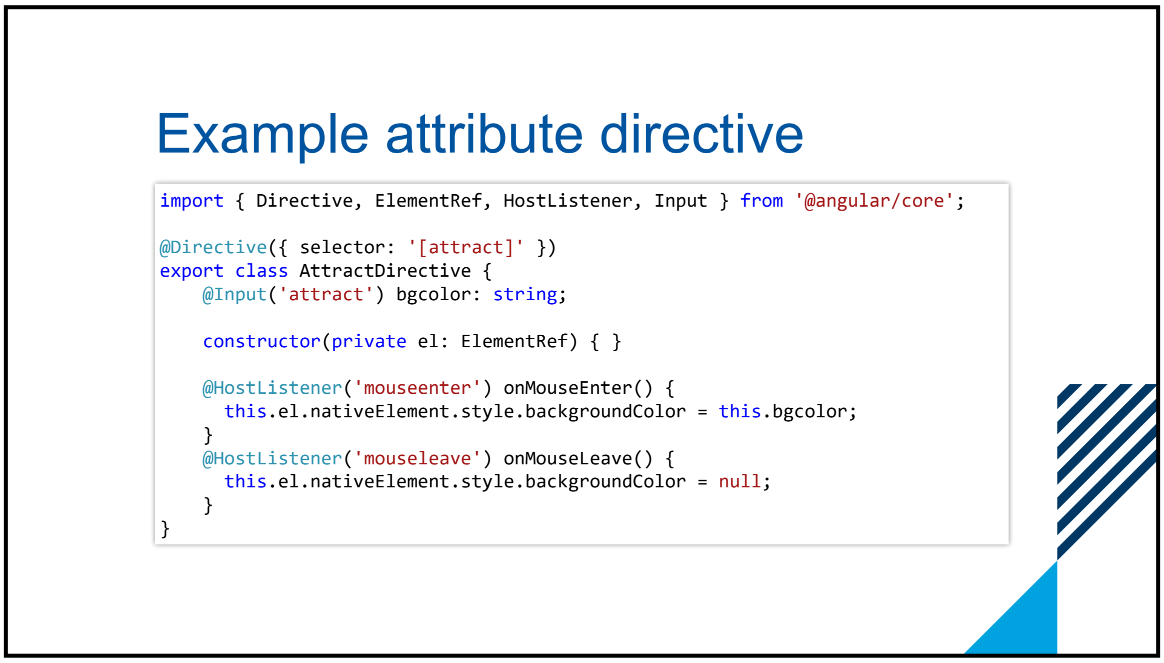


Attribute directives

Attribute directives change the appearance or behavior of an element, e.g. `ngStyle` and `ngClass`

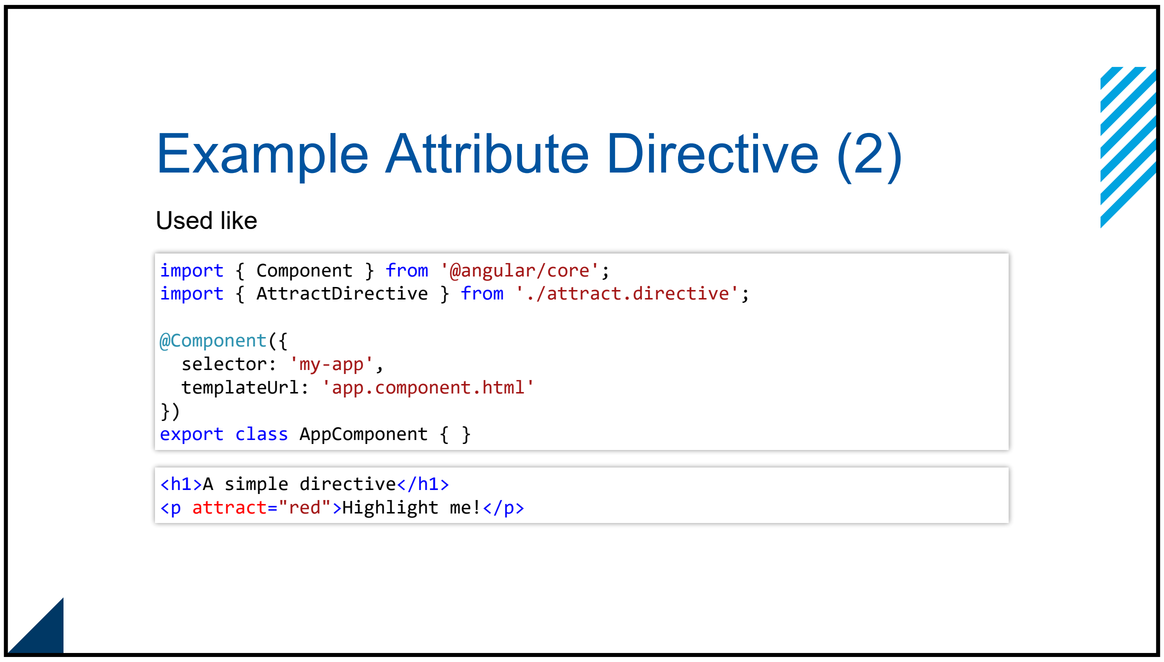
Note however:

- selector qualifier is set to '[myattribute]'
- Use `@HostListener()` to register for events
- `ElementRef` can be dependency injected to access the actual DOM element
- `@Input()` is used to set values



Example attribute directive

```typescript
import { Directive, ElementRef, HostListener, Input } from '@angular/core';

@Directive({ selector: '[attract]' })
export class AttractDirective {
    @Input('attract') bgcolor: string;

    constructor(private el: ElementRef) { }

    @HostListener('mouseenter') onMouseEnter() {
      this.el.nativeElement.style.backgroundColor = this.bgcolor;
    }
    @HostListener('mouseleave') onMouseLeave() {
      this.el.nativeElement.style.backgroundColor = null;
    }
}
```

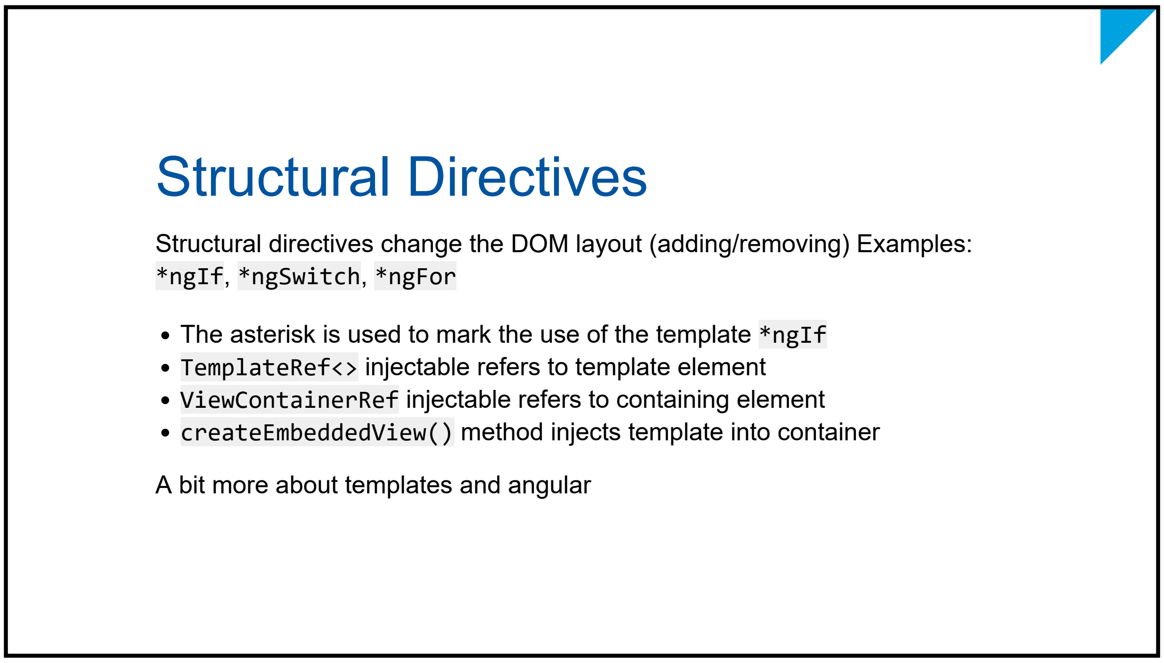


Example Attribute Directive (2)

Used like

```typescript
import { Component } from '@angular/core';
import { AttractDirective } from './attract.directive';

@Component({
  selector: 'my-app',
  templateUrl: 'app.component.html'
})
export class AppComponent { }
```

```html
<h1>A simple directive</h1>
<p attract="red">Highlight me!</p>
```

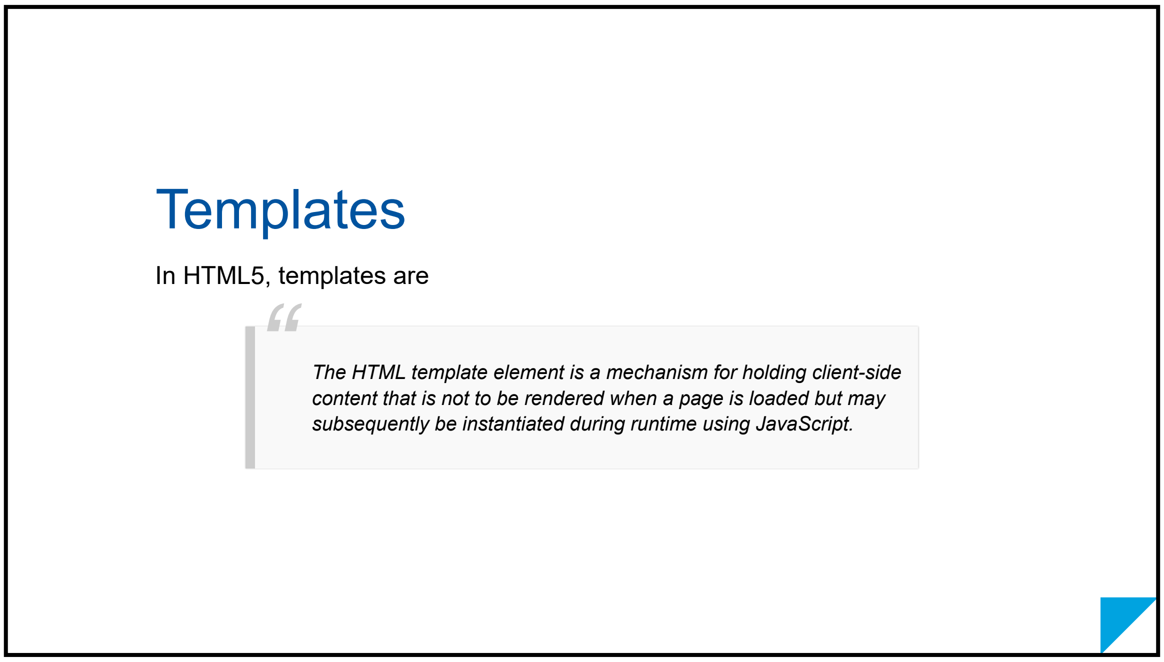


Structural Directives

Structural directives change the DOM layout (adding/removing) Examples: `*ngIf`, `*ngSwitch`, `*ngFor`

- The asterisk is used to mark the use of the template `*ngIf`
- `TemplateRef<>` injectable refers to template element
- `ViewContainerRef` injectable refers to containing element
- `createEmbeddedView()` method injects template into container

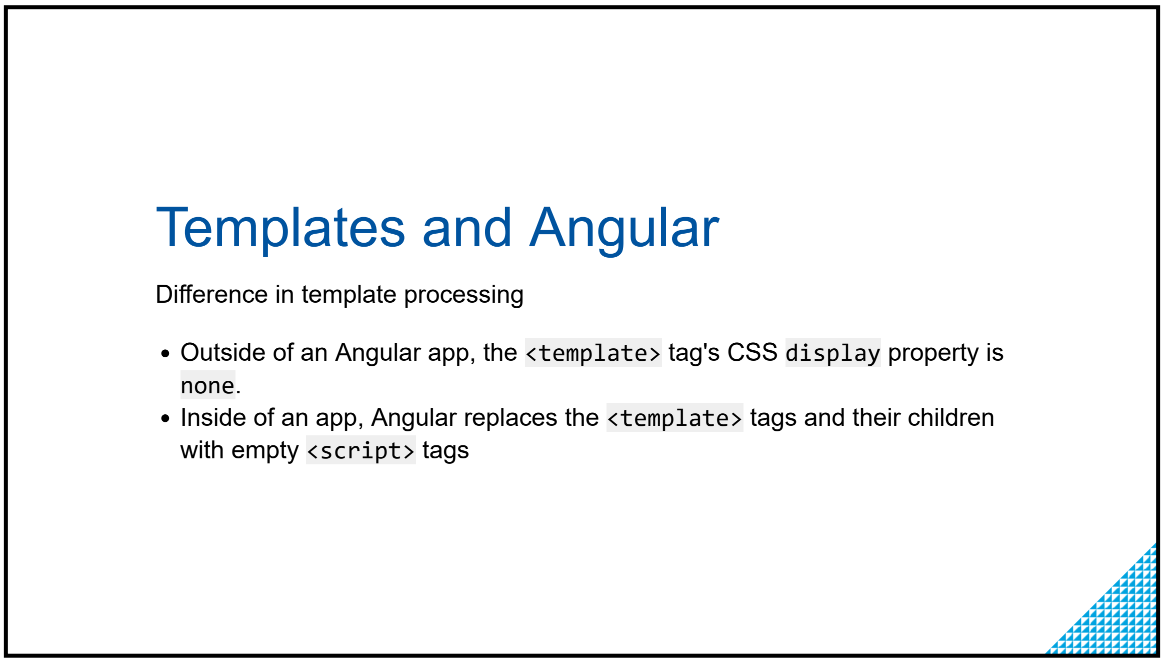
A bit more about templates and angular



Templates

In HTML5, templates are

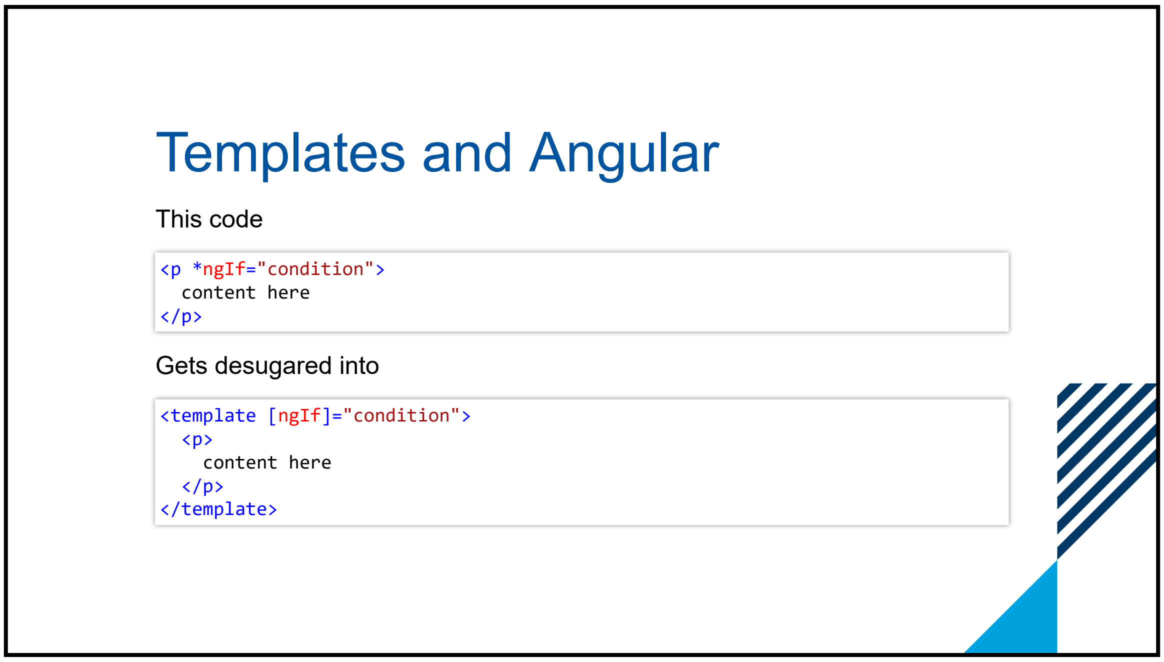
> *The HTML template element is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.*



Templates and Angular

Difference in template processing

- Outside of an Angular app, the `<template>` tag's CSS `display` property is `none`.
- Inside of an app, Angular replaces the `<template>` tags and their children with empty `<script>` tags



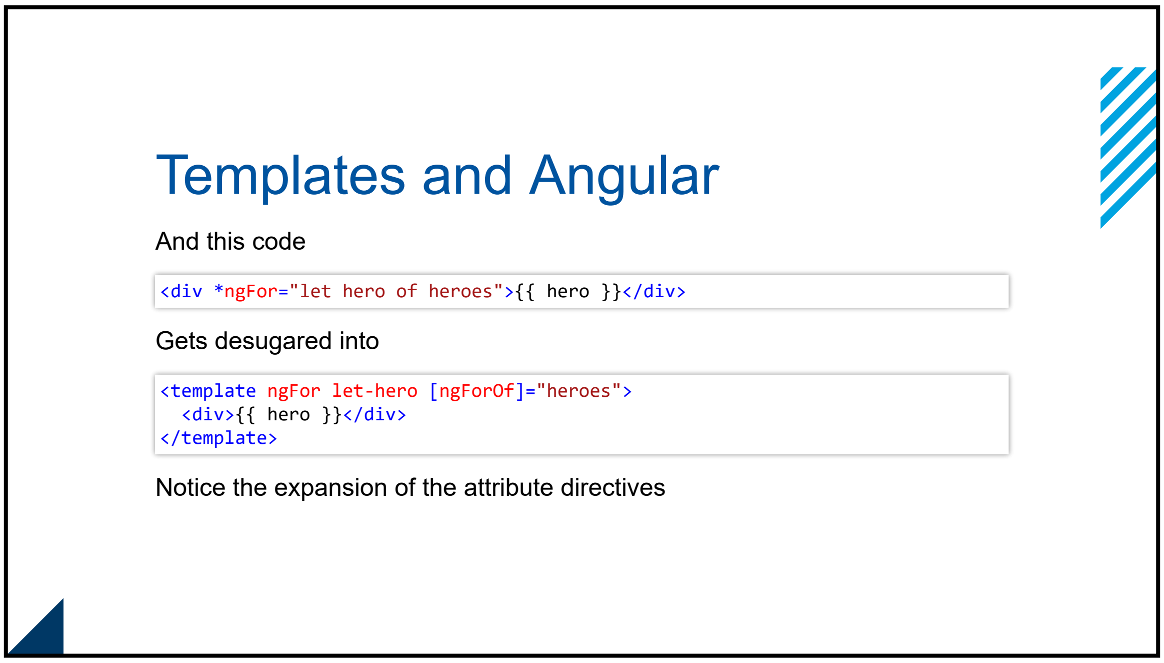
Templates and Angular

This code

```
<p *ngIf="condition">
  content here
</p>
```

Gets desugared into

```
<template [ngIf]="condition">
  <p>
    content here
  </p>
</template>
```



Templates and Angular

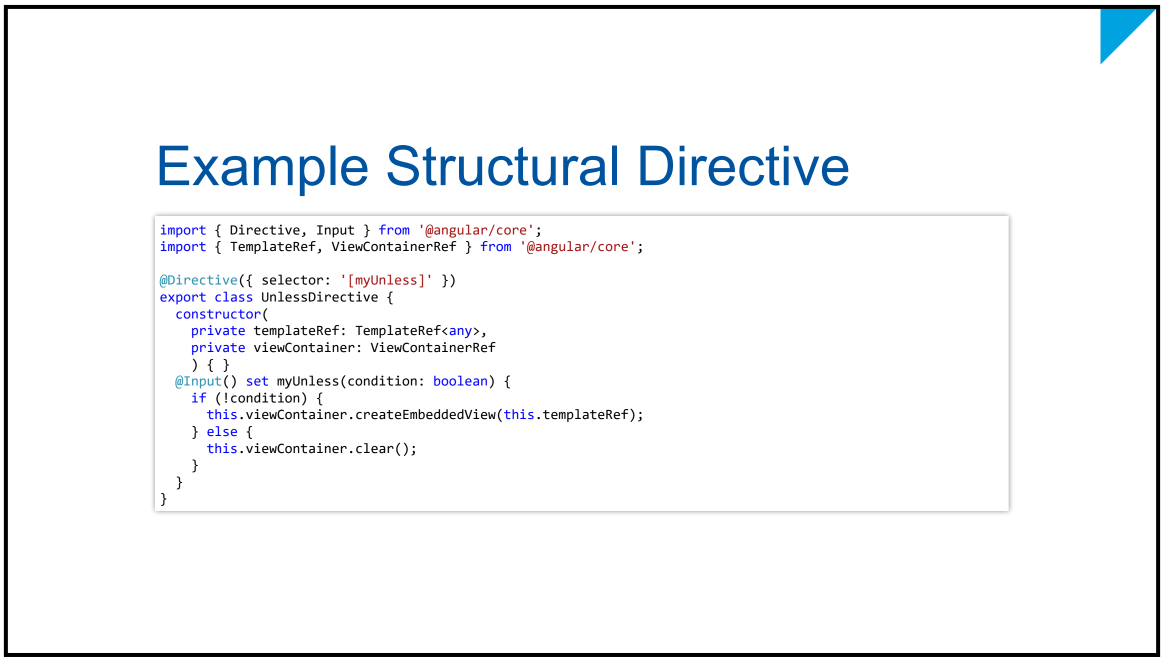
And this code

```
<div *ngFor="let hero of heroes">{{ hero }}</div>
```

Gets desugared into

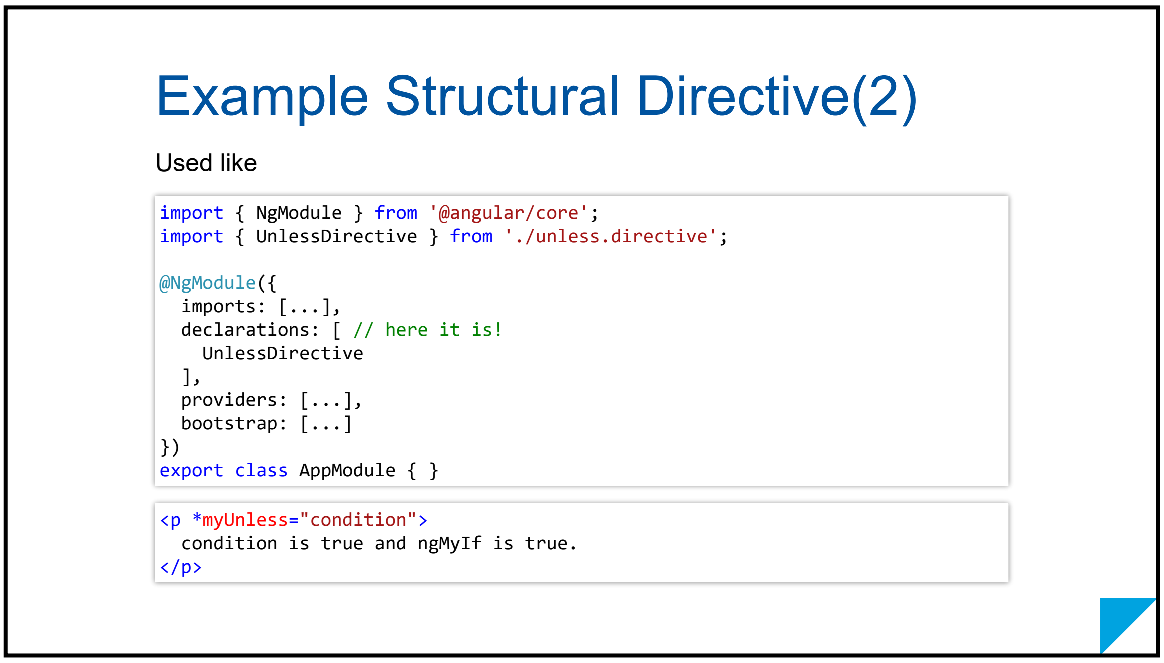
```
<template ngFor let-hero [ngForOf]="heroes">
  <div>{{ hero }}</div>
</template>
```

Notice the expansion of the attribute directives



Example Structural Directive

```
import { Directive, Input } from '@angular/core';
import { TemplateRef, ViewContainerRef } from '@angular/core';

@Directive({ selector: '[myUnless]' })
export class UnlessDirective {
  constructor(
    private templateRef: TemplateRef<any>,
    private viewContainer: ViewContainerRef
    ) { }
  @Input() set myUnless(condition: boolean) {
    if (!condition) {
      this.viewContainer.createEmbeddedView(this.templateRef);
    } else {
      this.viewContainer.clear();
    }
  }
}
```

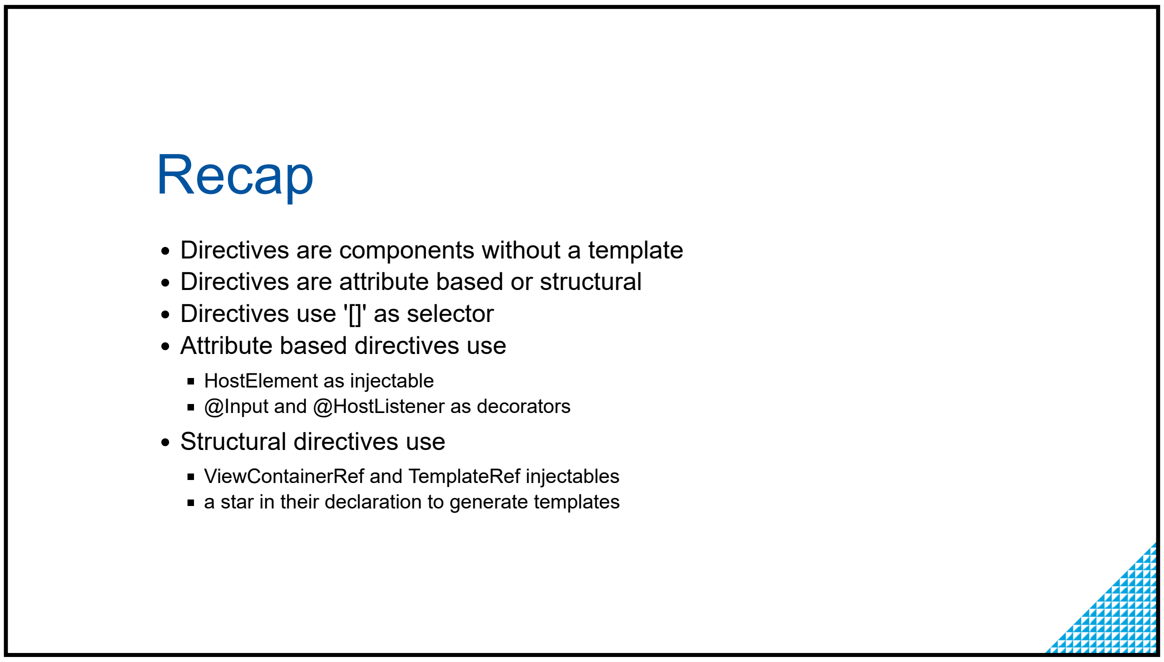


Example Structural Directive(2)

Used like

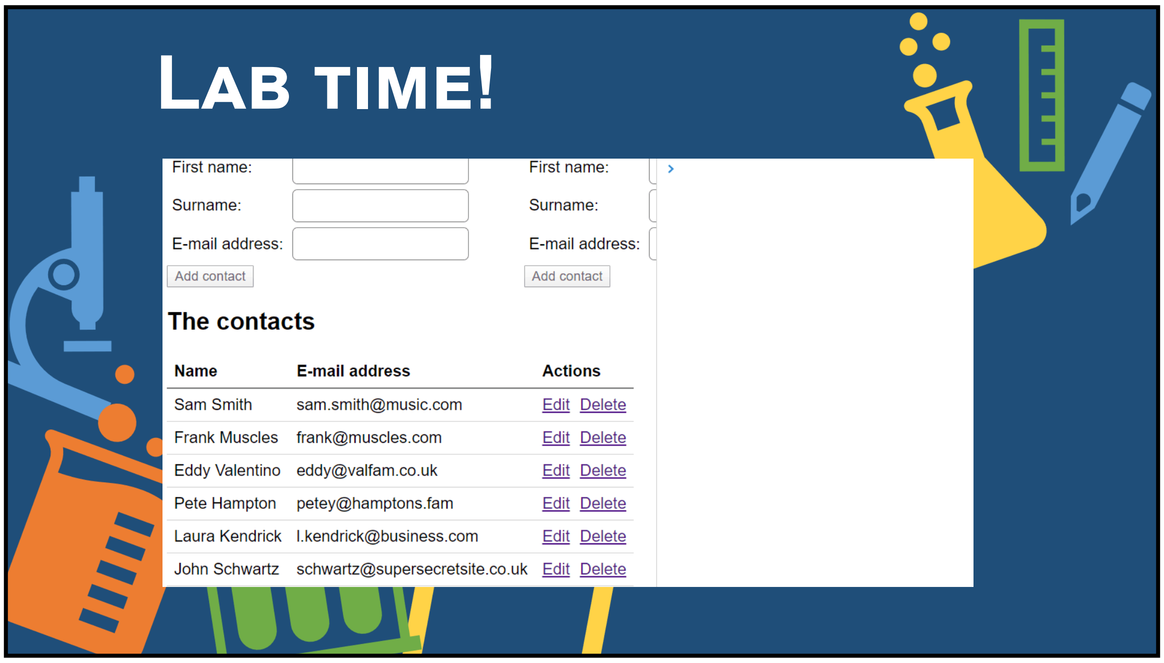
```
import { NgModule } from '@angular/core';
import { UnlessDirective } from './unless.directive';

@NgModule({
  imports: [...],
  declarations: [ // here it is!
    UnlessDirective
  ],
  providers: [...],
  bootstrap: [...]
})
export class AppModule { }
```

```
<p *myUnless="condition">
  condition is true and ngMyIf is true.
</p>
```



Recap

- Directives are components without a template
- Directives are attribute based or structural
- Directives use '[]' as selector
- Attribute based directives use
  - HostElement as injectable
  - @Input and @HostListener as decorators
- Structural directives use
  - ViewContainerRef and TemplateRef injectables
  - a star in their declaration to generate templates



Lab time!

First name:	
Surname:	
E-mail address:	

Add contact

First name:	
Surname:	
E-mail address:	

Add contact

The contacts

Name	E-mail address	Actions
Sam Smith	sam.smith@music.com	Edit Delete
Frank Muscles	frank@muscles.com	Edit Delete
Eddy Valentino	eddy@valfam.co.uk	Edit Delete
Pete Hampton	petey@hamptons.fam	Edit Delete
Laura Kendrick	l.kendrick@business.com	Edit Delete
John Schwartz	schwartz@supersecretsite.co.uk	Edit Delete