

Angular Training

# Directives

Peter Bouda, [hey@peterbouda.eu](mailto:hey@peterbouda.eu)



ng-lisbon.com

# What is a directive?

- Components without a view
- We use them to react on DOM events and to manipulate the DOM
- There are two types
  - Attribute directives
  - Structural directives
- Angular contains built-in directives for your app



ng-lisbon.com

# Attribute Directives

- Modify the appearance or the behaviour of a DOM element
- Are used as an attribute of an element
- Do not depend of a specific implementation of a component



ng-lisbon.com

# NgStyle

```
@Component({
  selector: 'app-style-example',
  template: `
    <p style="padding: 1rem"
      [ngStyle]="{
        'color': 'red',
        'font-weight': 'bold',
        'borderBottom': borderStyle
      }">
      <ng-content></ng-content>
    </p>
  `,
})
export class StyleExampleComponent {
  borderStyle = '1px solid black';
}
```

# NgClass

```
@Component({
  selector: 'app-class-as-string',
  template: `
    <p ngClass="centered-text underlined" class="orange">
      <ng-content></ng-content>
    </p>
  `,
  styles: [
    .centered-text {
      text-align: center;
    }
    .underlined {
      border-bottom: 1px solid #ccc;
    }
    .orange {
      color: orange;
    }
  ]
})
export class ClassAsStringComponent {
}
```

# Structural Directives

- Make use of the `template` element to specify how to render a component
- Special syntax as syntactic sugar

```
*structuralDirective="expression"
```

equivalent to:

```
<template [structuralDirective]="expression">  
...  
</template>
```

# Structural Directives

```
@Component({
  selector: 'app-directive-example',
  template: `
    <p *structuralDirective="expression">
      Under a structural directive
    </p>
  `,
})
```

Is equivalent to:

```
@Component({
  selector: 'app-directive-example',
  template: `
    <template [structuralDirective]="expression">
      <p>
        Under a structural directive
      </p>
    </template>
  `,
})
```

# NgIf

```
@Component({
  selector: 'app-root',
  template: `
    <button type="button" (click)="toggleExists()">
      Toggle Component
    </button>
    <hr>
    <app-if-example *ngIf="exists">
      Hello
    </app-if-example>
  `
})
export class AppComponent {
  exists = true;
  toggleExists() {
    this.exists = !this.exists;
  }
}
```



# NgFor

```
@Component({
  selector: 'app-root',
  template: `
    <app-for-example *ngFor="let episode of episodes" [episode]="episode">
      {{episode.title}}
    </app-for-example>
  `
})
export class AppComponent {
  episodes = [
    { title: 'Winter Is Coming', director: 'Tim Van Patten' },
    { title: 'The Kingsroad', director: 'Tim Van Patten' },
    { title: 'Lord Snow', director: 'Brian Kirk' },
  ];
}
```

# NgFor

## Local variables

index	The index of the current element
first	Is it the first element?
last	Is it the last element?
even	Is the current index even?
odd	Is the current index odd?



ng-lisbon.com

# NgFor

```
@Component({
  selector: 'app-root',
  template: `
    <app-for-example
      *ngFor="let episode of episodes; let i = index; let isOdd = odd"
      [episode]="episode"
      [ngClass]="{ odd: isOdd }">
        {{i+1}}. {{episode.title}}
      </app-for-example>
    `
})
```

# NgSwitch

```
@Component({
  selector: 'app-root',
  template: `
    <div class="tabs-selection">
      <app-tab [active]="isSelected(1)" (click)="setTab(1)">Tab 1</app-tab>
      <app-tab [active]="isSelected(2)" (click)="setTab(2)">Tab 2</app-tab>
    </div>
    <div [ngSwitch]="tab">
      <app-tab-content *ngSwitchCase="1">Tab content 1</app-tab-content>
      <app-tab-content *ngSwitchCase="2">Tab content 2</app-tab-content>
      <app-tab-content *ngSwitchDefault>Select a tab</app-tab-content>
    </div>
  `
})
export class AppComponent {
  tab: number = 0;
  setTab(num: number) { this.tab = num; }
  isSelected(num: number) { return this.tab === num; }
}
```

# Custom attribute directive

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

@Directive({
  selector: '[appChangeColor]'
})
export class ChangeColorDirective {
  @Input() defaultColor = 'green';
  @HostBinding('style.color') color = this.defaultColor;

  @HostListener('mouseenter') mouseenter() {
    this.color = 'red';
  }

  constructor() { }
}
```

In HTML:

```
<li appChangeColor [defaultColor]="green">{{ menuItem }}</li>
```

# Custom structural directive

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

@Directive({
  selector: '[appShowActive]'
})
export class ShowActiveDirective {
  @Input() set appShowActive(item) {
    if(item.active) {
      this.viewContainer.createEmbeddedView(this.templateRef);
    }
  }
  constructor(private templateRef: TemplateRef<any>,
    private viewContainer: ViewContainerRef) {}
}
```

In HTML:

```
<li *ngFor="let menuItem of menuItems">
  <span *appShowActive="menuItem">{{ menuItem.title }}</span>
</li>
```