



Directives

Fravezzi Mattia
m.fravezzi@almaviva.it

Directives

1. **Components**—directives with a template. This type of directive is the most common directive type.
1. **Attribute directives**—directives that change the appearance or behavior of an element, component, or another directive.
1. **Structural directives**—directives that change the DOM layout by adding and removing DOM elements.

Attribute Directives

- `NgClass`—adds and removes a set of CSS classes.
- `NgStyle`—adds and removes a set of HTML styles.
- `NgModel`—adds two-way data binding to an HTML form element.

NgClass

```
<div [ngClass]="isSelected ? 'isSelected' : 'isSelected'" data-bbox="82 338 625 454">></div>

<div [ngClass]="{'isSelected': isSelected}" data-bbox="82 425 625 454">></div>
```

```
public currentClasses = {
  saveable: true,
  modified: false,
  isSelected: true
};

<div [ngClass]="currentClasses" data-bbox="82 875 393 905">></div>
```

NgModel

```
<input MatInput [(ngModel)]="heroName">
<button (click)="saveHero()" >Save</button>
```

```
public heroList: string[] = [];
public heroName: string = '';

public saveHero() {
    this.heroList.push(this.heroName);
    this.heroName = '';
}
```

```
@NgModule({
  declarations: [
    ...
  ],
  imports: [
    ...
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

Custom Directives

ng g directive

```
@Directive({  
  selector: '[appHighlight]'  
})  
export class HighlightDirective {  
  
  constructor(el: ElementRef) {  
    el.nativeElement.style.backgroundColor = 'yellow';  
  }  
}
```

Custom Directives - Handling user events

```
export class HighlightDirective {  
  @Input()  
  public color: string = 'yellow';  
  
  @HostListener('mouseenter') onMouseEnter() {  
    this.highlight(this.color);  
  }  
  @HostListener('mouseleave') onMouseLeave() {  
    this.highlight('');  
  }  
  private highlight(color: string) {  
    this.element.nativeElement.style.backgroundColor = color;  
  }  
  constructor(  
    private element: ElementRef  
  ) { }  
}
```

Structural Directives

- `*ngIf`
- `*ngFor`
- `*ngSwitch`

Custom Directives

```
export class UnlessDirective {
  @Input() set appUnless(condition: boolean) {
    if (!condition && !this.hasView) {
      this.viewContainer.createEmbeddedView(this.templateRef);
      this.hasView = true;
    } else if (condition && this.hasView) {
      this.viewContainer.clear();
      this.hasView = false;
    }
  }
  private hasView = false;
  constructor(
    private templateRef: TemplateRef<any>,
    private viewContainer: ViewContainerRef
  ) { }
}
```

```
<p *appUnless="condition">
  Show (A).
</p>
```

```
<p *appUnless="!condition" >
  Show (B).
</p>
```