



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Directives Deep Dive

Alex Thalhammer

Contents

- Attribute Directives
- Templates and View Containers
- Structural Directives

Directives

- „Components without Templates“
- Add behavior to an element
- Examples: ngModel, ngClass, ngStyle

Attribute Directives

Case Study

```
<button appClickWithWarning>Delete</button>
```

Simple Example

```
@Directive({
  selector: '[appClickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {
  constructor(private elementRef: ElementRef) {}

  ngOnInit(): void {
    this.elementRef
      .nativeElement.setAttribute('class', 'btn btn-danger');
  }
}
```

Calling a Directive

```
<button appClickWithWarning>Delete</button>
```



Host-Element

Bindings

```
@Directive({
  selector: '[appClickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {
  @Input() warning = 'Are you sure?';
  @Output() appClickWithWarning = new EventEmitter<void>();
}
```

Bindings

```
@Directive({
  selector: '[appClickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {
  @Input() warning = 'Are you sure?';
  @Output() appClickWithWarning = new EventEmitter();

  @HostBinding('class') classBinding = 'btn btn-danger';
}
```

Bindings

```
@Directive({
  selector: '[appClickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {
  @Input() warning = 'Are you sure?';
  @Output() appClickWithWarning = new EventEmitter<void>();

  @HostBinding('class') classBinding = 'btn btn-danger';

  @HostListener('click', ['$event'])
  handleClick($event: MouseEvent): void { ... }
}
```

Calling the Directive

```
<button (appClickWithWarning)="delete()" warning="Sure?>Delete</button>
```

DEMO

Templates and ViewContainers

Example: Tooltip

```
<input [appTooltip]="tmpl"> <- - ViewContainer  
  
<ng-template #tmpl>  
  <h3>2 Tips for Success</h3>  
  <ol>  
    <li>Don't tell everything!</li>  
  </ol>  
</ng-template>
```

Implementation

```
@Directive({
  selector: '[appTooltip]'
})
export class TooltipDirective implements OnInit {
  @Input('appTooltip') template: TemplateRef<unknown>;
  constructor(private host: ElementRef,
    private viewContainerRef: ViewContainerRef) { }

  ngOnInit(): void {
    this.viewContainerRef.createEmbeddedView(this.template);
  }
}
```

Implementation

```
export class TooltipDirective implements OnInit {  
  @Input('appTooltip') template: TemplateRef<unknown>;  
  private viewRef: EmbeddedViewRef<unknown>;  
  
  constructor(  
    private host: ElementRef,  
    private viewContainer: ViewContainerRef) {}  
  
  ngOnInit(): void {  
    this.viewRef = this.viewContainer.createEmbeddedView(this.template);  
    [...]  
  }  
}
```

Mouse-Events

```
const elm = this.host.nativeElement as HTMLElement;  
  
elm.addEventListener('mouseover', () => {  
  [...]  
});  
  
elm.addEventListener('mouseout', () => {  
  [...]  
});
```

Iterate over Projected Root Nodes

```
this.viewRef.rootNodes.forEach(nativeElement => {
  nativeElement.hidden = true;
});
```

```
<input [appTooltip]="tmpl">

<ng-template #tmpl>
  <h3>2 Tips for Success</h3>← = = - - Root Nodes
  <ol>← - - - -
    <li>Don't tell everything!</li>
  </ol>
</ng-template>
```

DEMO

Adding Templates and Components on the Fly

ViewContainerRef

- createEmbeddedView
- createComponent

TemplateOutletDirective

Insert Template in Placeholder

```
<div *ngTemplateOutlet="tmpl"></div>
<ng-template #tmpl>Hello world!</ng-template>
```

ComponentOutletDirective

Insert Component in Placeholder

```
<div *ngComponentOutlet="FlightSearchComponent"></div>
```

ContentChildren

- **@ContentChildren(MyComponentOrDirective,
{ read: ElementRef | ViewContainerRef })**
- Same for @ContentChild, @ViewChildren, @ViewChild

DEMO

LAB

Structural Directives

Structural Directive

Micro Syntax

```
<div *ngFor="let f of flights; index as i">  
  <pre>{{i}}: {{ f | json }}</pre>  
</div>
```

Template

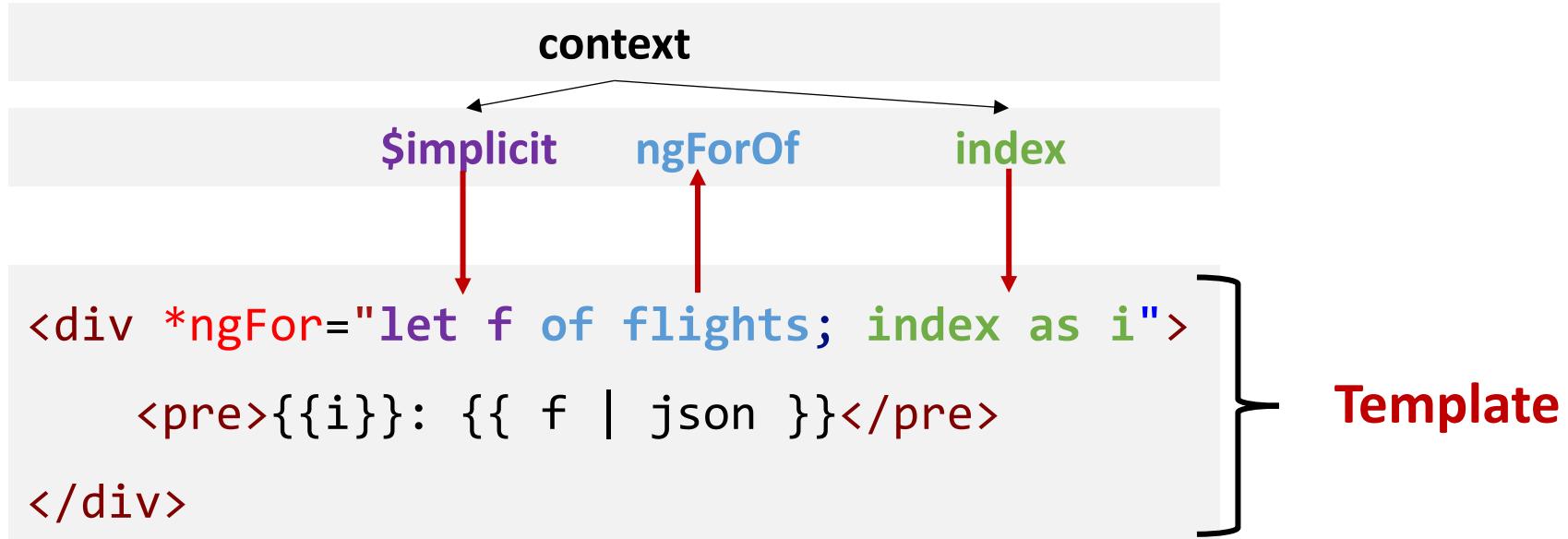
Structural Directive

ngFor Implementation

```
<div *ngFor="let f of flights; index as i">
  <pre>{{i}}: {{ f | json }}</pre>
</div>
```

Template

Structural Directive



Syntax Sugar

```
<div *ngFor="let f of flights; index as i">  
  <pre>{{i}}: {{ f | json }}</pre>  
</div>
```

```
<ng-template ngFor let-f [ngForOf]="flights" let-i="index">  
  <div>  
    <pre>{{i}}: {{ f | json }}</pre>  
  </div>  
</ng-template>
```

Case Study #2: DataTable

Upcoming Flights

1	Hamburg	Berlin	01.02.2025 17:00
2	Hamburg	Frankfurt	01.02.2025 17:30
3	Hamburg	Mallorca	01.02.2025 17:45

DataTable

```
<app-data-table [data]="flights">
  <div *appTableField="let data as 'id'">{{data}}</div>
  <div *appTableField="let data as 'from'">{{data}}</div>
  <div *appTableField="let data as 'to'">{{data}}</div>
  <div *appTableField="let data as 'date'">
    {{data | date:'dd.MM.yyyy HH:mm'}}</div>
</app-data-table>
```

DEMO

LAB

Summary

- Attribute Directive with Input, Output, HostBinding, HostListener
- ElementRef, TemplateRef
- *ngTemplateOutlet, *ngComponentOutlet
- ViewContainerRef, EmbeddedViewRef
- Structural Directive: Template + Micro Syntax