

3 Arten von Direktiven

Direktiven

Manipulieren die HTML durch Einfügen/Entfernen von DOM Elementen oder Änderung von Attributen.

Strukturelle Direktiven

- Bestimmen das HTML-Layout bzw. die DOM-Struktur
- 1:1-Zuordnung zu Hostelement

Attribut Direktiven

- Verändert Erscheinung und/oder Verhalten von Elementen, Komponenten oder Direktiven
- n:1-Zuordnung zu Hostelement

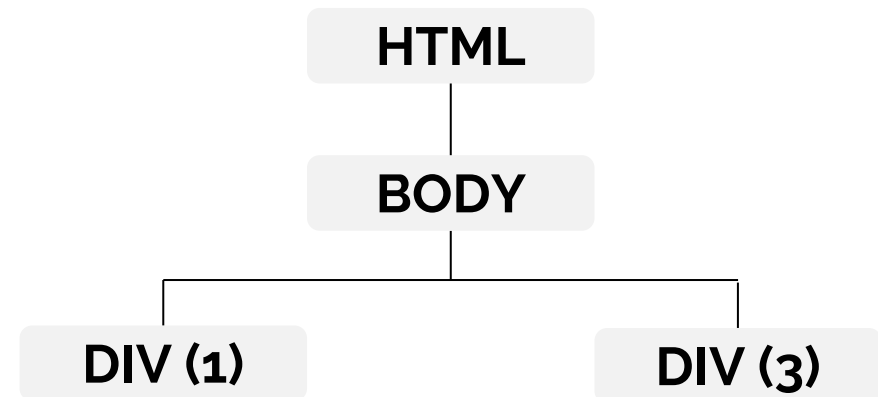
Komponenten

- Verwaltet eine Ansicht
- Werden keinen Elementen zugewiesen, sondern besitzen ein eigenes Template

DOM-Manipulation

- Verantwortlich für das HTML-Layout. DOM-Elemente können hinzugefügt, entfernt oder geändert werden.
- Müssen einem Host-Element zugewiesen werden.
- Kennzeichnung mittels des Stern-Zeichens (*)

```
<html>  
  <body>  
    <div>1</div>  
    <div *ngIf="false">2</div>  
    <div>3</div>  
  </body>  
</html>
```



DOM-Manipulation

If:

```
<div *ngIf="<bedingung>" ; else login">  
  Der Benutzer ist eingeloggt.  
</div>  
<ng-template #login>Bitte einloggen.</ng-template>
```

If-Then-Else:

```
<div *ngIf="<bedingung>; then logout else login"></div>  
  
<ng-template #login>Bitte einloggen.</ng-template>  
<ng-template #logout>Willkommen! <button>Logout</button></ng-template>
```

#login und #logout sind Template Referenzvariablen. Diese ermöglichen DOM-Zugriffe des Hostelements.

DOM-Manipulation

NgFor

```
let heroes = [  
  new Hero("Batman", "happy"),  
  new Hero("Wonder Woman", "happy")  
]
```

```
<ul>  
  <li *ngFor="let hero of heroes">{{hero.name}}</li>  
</ul>
```

NgSwitch

```
<div [ngSwitch]="hero.emotion">  
  <happy-hero *ngSwitchCase="'happy'" [hero]="hero"></happy-hero>  
  <sad-hero *ngSwitchCase="'sad'" [hero]="hero"></sad-hero>  
  <unknown-hero *ngSwitchDefault [hero]="hero"></unknown-hero>  
</div>
```

DOM-Manipulation

Unterschied zwischen Entfernen und Ausblenden:

- Beim Ausblenden bleiben Elemente im DOM erhalten
- Ausgeblendete Elemente geben keine Ressourcen frei. Prozesse laufen im Hintergrund weiter:
 - Event-Listeners
 - Change-Detection
 - Interne Logik (Websockets, etc.)
- Entfernen und wieder Erstellen von Elementen kann Performance intensiv sein (Neuinitialisierung der Komponente).

Mehrere strukturelle Direktiven

Ein Hostelement darf nur eine strukturelle Direktive besitzen:

```
<table>  
  <tr *ngFor="let entity of entities" *ngIf="entities">  
    <td>{{entity.title}}</td>  
  </tr>  
</table>
```



```
<table *ngIf="entities">  
  <tr *ngFor="let entity of entities">  
    <td>{{entity.title}}</td>  
  </tr>  
</table>
```



Hintergrund: Stern-Zeichen (*)

„Syntactic sugar“ für interne Angular Logik. Zwei-Schritte Umwandlung:

Entwickler Schreibweise:

```
<table *ngIf="entities">...</table>
```

1. Umwandlung in Template-Attribut:

```
<table template="ngIf entities">...</table>
```

2. Umwandlung in Template-Element (Wrapppper):

```
<template [ngIf]="entities">  
  <table>...</table>  
</template>
```

NgContainer

Verwendung von NgContainer, sofern kein Elternelement vorhanden ist:

```
<p>
```

```
I turned the corner <ng-container *ngIf="hero"> and saw  
{{hero.name}}. I waved </ng-container> and continued on  
my way.
```

```
</p>
```

Vorteil gegenüber Wrapper-Element:

- Vermeidung von CSS-Konflikten
- Keine Verletzung von direkten Kindelementen (select -> option)

Attribut-Direktiven

Grundgerüst einer Attribut-Direktive:

```
import { Directive, ElementRef, Input } from '@angular/core';
@Directive({ selector: '[myHighlight]' })
export class HighlightDirective {
  constructor(el: ElementRef) {
    el.nativeElement.style.backgroundColor = 'yellow';
  }
}
```

```
<p myHighlight>Highlight me!</p>
```

Zusammenfassung

Um was ging es in diesem Modul?

- Manipulation der Browser DOM durch Einfügen und Entfernen von Elementen
- Änderung von HTML Elementattributen
- Entfernen / Ausblenden Performanceimplikationen

Wozu brauche ich das? Was werde ich damit machen?

- Konditionell Daten anzeigen
- HTML Listen ausgeben

Kontrollfragen

- Wie viele Arten Direktiven stellt Angular zur Verfügung?
- Was sind die Unterschiede zwischen diesen?
- Wie kann ich Information nicht mehr im Browser anzeigen, und welche Auswirkungen hat das auf Performance?
- Warum kann ich nicht mehr als eine Direktive pro Element angeben?

