



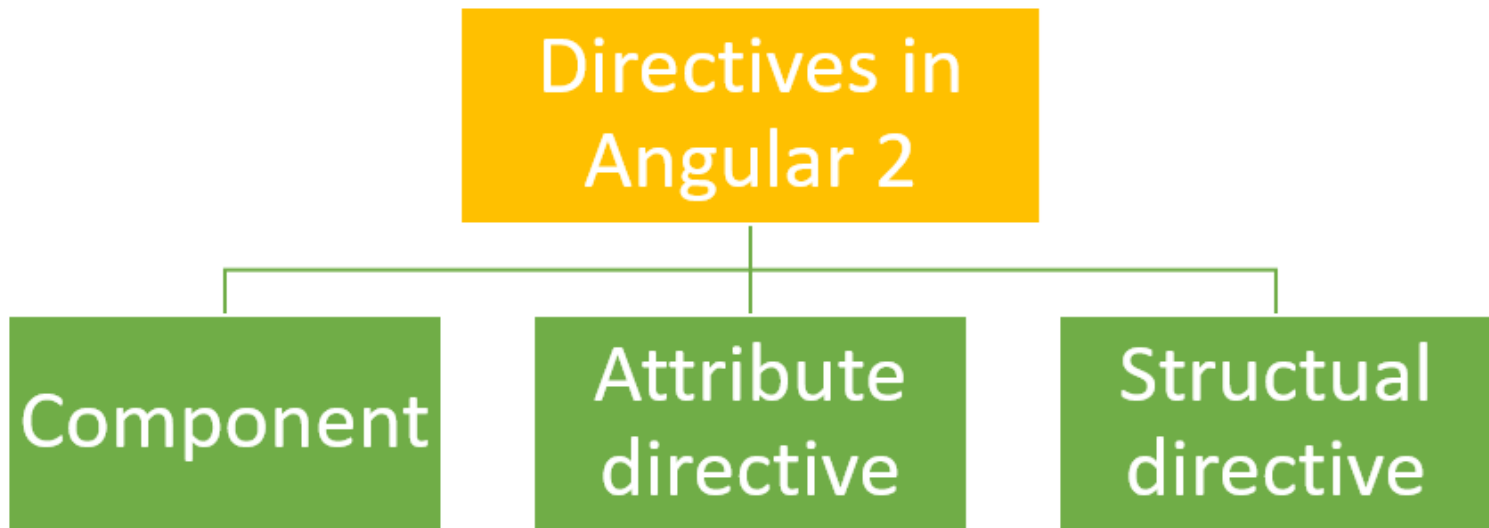
# Directives



# Agenda

- Атрибутивні директиви
- Структурні директиви
- HostListener і HostBinding
- ng-container, ng-template

# Види директив



- Компоненти - підвид директив з шаблоном
- Атрибутивні - змінюють поведінку вже існуючого елемента, до якого вони застосовуються
- Структурні - змінюють структуру DOM за допомогою додавання, зміни або видалення елементів html.

# Існуючі атрибутивні директиви

- `ngClass` дозволяє визначити набір класів, які будуть застосовуватися до елемента

```
<p [ngClass]="{'blue': color==true, 'yellow': color==false}">
```

```
  Text
```

```
</p>
```

- `ngStyle` дозволяє задати набір стилів, які застосовуються до елемента

```
<div [style.font-size]="isSpecial ? 'x-large' : 'smaller'" >
```

```
  This div is x-large or smaller.
```

```
</div>
```

# Створення атрибутивних директив

- створення директиви(cli) `ng g d <name>`

```
import { Directive } from '@angular/core';
```

```
@Directive({
```

```
  selector: '[appColor]'
```

```
})
```

```
export class ColorDirective {
```

```
  constructor() { }
```

```
}
```

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

@Directive({
  selector: '[appColor]' // селектор за допомогою якого можна застосовувати директиву
})

export class ColorDirective {
  @Input() highlightColor: string;

  constructor(
    private el: ElementRef
  ) {
    el.nativeElement.style.color = this.highlightColor; // директива має доступ до DOM, до якого
вона застосована
  }
}

<p appColor [highlightColor]='orange'> // виклик директиви через селектор, може застосовуватися і
до компонентів

  Start editing to see some magic happen :)

</p>
```

[Demo](#)

# Структурні директиви

\* (зірочка) вказує, що це структурна директива

- \*ngIf - дозволяє видалити або, навпаки, відобразити елемент за певної умови
- \*ngFor - дозволяє перебрати в шаблоні елементи масиву
- \*ngSwitch - можна вбудувати в шаблон конструкцію switch..case і в залежності від її результату виконання виводити той чи інший блок



```
<div *ngIf="hero" class="name">{{hero.name}}</div>
```

```
<ul>
```

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

```
</ul>
```

```
<div [ngSwitch]="emotion">
```

```
  <app-happy-hero      *ngSwitchCase="'happy' "      [hero]="hero"></app-happy-hero>
```

```
  <app-sad-hero        *ngSwitchCase="'sad' "         [hero]="hero"></app-sad-hero>
```

```
  <app-confused-hero   *ngSwitchCase="'confused' "    [hero]="hero"></app-confused-hero>
```

```
  <app-unknown-hero    *ngSwitchDefault              [hero]="hero"></app-unknown-hero>
```

```
</div>
```

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

@Directive({
  selector: '[appDelay]'
})
export class DelayDirective {
  constructor(
    private template: TemplateRef<any>, // посилається на елемент до якого застосована директива
    private view: ViewContainerRef      // обгортка навколо директиви
  ) { }

  ngOnInit() {
    setTimeout( _=> {
      this.view.createEmbeddedView(this.template) // в контейнер додаємо шаблон
    }, 2000);
  }
}
```

[Demo](#)

# HostListener і HostBinding

HostBinding - дозволяє зв'язати звичайне властивість класу з властивістю елемента, до якого застосовується директива

```
@HostBinding('style.color') myColor;  
  
this.myColor = 'red'
```

HostListener - дозволяє зв'язати події DOM і методи директиви

```
@HostListener('click') onClick() {  
    console.log('test');  
};
```

# <ng-template>

- ніколи не відобразиться без задання структурної директиви

```
<p>Hip!</p>  
<ng-template>  
  <p>Hip!</p>  
</ng-template>  
<p>Hooray!</p>
```

```
<p _ngcontent-c0>Hip!</p>  
<!-- -->  
<p _ngcontent-c0>Hooray!</p>
```

Hip!

Hooray!

# <ng-container>

- застосовується коли не потрібно задавати додатковий html щоб застосувати структурну директиву

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

нерозриває рядок

I turned the corner and saw Mr. Nice. I waved and continued on my way.

# Посилання

<https://angular.io/guide/attribute-directives>

<https://metanit.com/web/angular2/3.2.php>

<https://www.sitepoint.com/practical-guide-angular-directives/>

<https://codecraft.tv/courses/angular/custom-directives/creating-a-custom-directive/>

<https://www.edureka.co/blog/angular-directive/#Directives>

# Exemple

<https://stackblitz.com/edit/angular-tooltip-directive> - директива tooltip(підказки на ховер)

<https://stackblitz.com/edit/angular-numbers-only-directive> - input тільки числа

<https://stackblitz.com/edit/angular-6-custom-structural-directive> - структурна директива

# Відео

<https://learn.javascript.ru/screencast/angular#components-attr-directive>

[https://www.youtube.com/watch?v=HD-6QsvpfeM&list=PLVfMKQXDAhGW12JY3SfeDnEx7S5\\_tn11j&index=4](https://www.youtube.com/watch?v=HD-6QsvpfeM&list=PLVfMKQXDAhGW12JY3SfeDnEx7S5_tn11j&index=4)