

# Dyrektywy w Angular:

- Dyrektywy strukturalne – dodawanie / usuwanie elementów z DOM.
  - \*ngFor
  - \*ngIf
  - \*ngSwitch
- Dyrektywy atrybutowe – zmiana wyglądu elementów DOM.
  - ngStyle
  - ngClass
  - ngModel
- Komponenty

Komponent to także dyrektywa, ale posiada tzw. widok definiowany przez szablon HTML.

Oprócz wbudowanych dyrektyw możemy tworzyć także swoje własne dyrektywy.

## Dyrektywa \*ngFor:

- Pozwala iterować po kolekcji danych np. po tablicy i wyświetlić odpowiednie dane. Poniżej przykład.

```
8  export class AppComponent {  
    no usages  
9  names = ['Alina', 'Bartek', 'Cecylia', 'Dariusz'];  
    1 usages
```

```
<> app.component.html ×  
1  <ul>  
2    <li *ngFor="let name of names">{{name}}</li>  
3  </ul>  
4
```

- Alina
- Bartek
- Cecylia
- Dariusz

Końcowy wynik w przeglądarce

# Dyrektywa \*ngIf

- Pozwala na dynamiczne dodawanie lub usuwanie elementów z interfejsu użytkownika w zależności od wartości warunku. Elementy są usuwane z drzewa DOM i nie można ich podglądać w zakładce **Elements** w devtools.

```
export class AppComponent {  
  1 usage  
  displaySomething = false;  
  1 usage
```

```
<> app.component.html × TS app.component.ts  
1 <p *ngIf="displaySomething">Wyświetlam element!</p>  
2
```

Powyższy element `<p>` nie zostanie wyświetlony w przeglądarce, ponieważ właściwość **displaySomething** w `app.component.ts` ma wartość **false**.

# Dyrektywa \*ngSwitch

- Pozwala na dynamiczne przełączanie pomiędzy różnymi elementami interfejsu użytkownika w zależności od wartości wyrażenia.

```
export class AppComponent {  
  1 usage  
  switchCase = 'test'  
  1 usage
```

```
<> app.component.html x app.component.ts  
1 <div [ngSwitch]="switchCase">  
2   <p *ngSwitchCase="'test'">Wyświetlam element!</p>  
3   <p *ngSwitchCase="'innyTest'">Wyświetlam drugi element!</p>  
4   <p *ngSwitchDefault>Wyświetlam default!</p>  
5 </div>
```

Wyświetlam element!

W przeglądarce wyświetli się zawartość pierwszego `<p>`, ponieważ właściwość **switchCase** ma taką samą wartość, jak warunek przy **\*ngSwitchCase**.

# ngStyle

- Pozwala na dynamiczne ustawianie stylów CSS na elemencie w oparciu o wartości właściwości w komponencie.

```
8   export class AppComponent {  
    1 usage  
9   isRed = false;
```

```
<> app.component.html × TS app.component.ts  
1 <p [ngStyle]="{'color': isRed ? 'red' : 'black'}">Kolor nadawany warunkowo!</p>  
2
```

Kolor nadawany warunkowo!

W tym przypadku kolorem pozostanie czarny, ponieważ właściwość **isRed** jest ustawiona na wartość **false**.

# ngClass

- Pozwala na dynamiczne dodawanie lub usuwanie klas CSS na elemencie w oparciu o wartości właściwości w komponentach.

```
8 export class AppComponent {  
  1 usage  
9   useClass = true;  
  1 usage
```

```
<> app.component.html × app.component.css app.component.ts  
1 <p [ngClass]='{'class1 class2': useClass}'>Klasa nadawana warunkowo!</p>  
2
```

```
1 .class1 {  
2   color: red;  
3 }
```

Klasa nadawana warunkowo!

Właścwość **useClass** ma wartość **true**, więc element **<p>** otrzyma dwie klasy – **class1** oraz **class2**.