

TypeScript

ANGULAR

Dyrektywy: dostęp do struktury dokumentu tak aby możliwa była zmiana wyglądu i zachowania danego elementu strony. Wbudowane dyrektywy mają w nazwie prefix **ng**. Dyrektywy pozwalają na szybsze działanie aplikacji ponieważ strona wczyta tylko te dane które są aktualnie potrzebne.

Dyrektywy w Angularze dzielimy na 3 kategorie:

- komponenty
- dyrektywy strukturalne
- dyrektywy atrybutowe

Do dzisiejszej lekcji potrzebujemy nowego projektu! Stwórz go – nazwa AngularDyrektywy (lub inna jeśli ta Ci się nie podoba).

Zmień **templateUrl** na plik o twoim imieniu i nazwisku.

Stwórz folder **interface** a w nim plik **osoba.ts**. Użyjemy go do stworzenia obiektu osoba z określonymi danymi. U mnie wygląda to tak: (**export** oznacza że można go importować w innym pliku)

```
angular_dyrektywy > src > app > interface > TS osoba.ts > Osoba
1  export interface Osoba {
2      name: String;
3      lastName: String;
4      picture: String;
5  }
```

W pliku **app.components.ts** importujemy powyższy plik jako **Osoba**.

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

import { Osoba } from './interface/osoba';

@Component({
  selector: 'app-root',
  templateUrl: './dyrektywy.html',
  styleUrls: ['./app.component.css']
})
```

Teraz możemy utworzyć nasz obiekt celem późniejszego użycia na stronie HTML. Poniższe dopisujemy do: **export class AppComponent {**

```
//Stworzenie obiektu i nadanie mu własności - pola mają wartość
osoba:Osoba = {
    name:"Jan",
    lastName: "Brzechwa",
    picture: "assets/images/osoba1.jpg" //folder assets/images/
}
```

Użycie na naszej stronie:

```
<p>ngIf</p>
<p>Imię: {{osoba.name}}</p>
<p>Nazwisko: {{osoba.lastName}}</p>

```

Wynik działania – wygląd strony HTML:

ngIf

Imię: Jan

Nazwisko: Brzechwa



Teraz możemy przejść do tworzenia **DYREKTYW**!

DYREKTYWY STRUKTURALNE: zmieniające kod HTML oraz jego układ

Dyrektywa ngIf – wybór na podstawie PRAWDA – FAŁSZ co wyświetlić

```
<p>ngIf</p>
<p>Imię: {{osoba.name}}</p>
<p>Nazwisko: {{osoba.lastName}}</p>
 <br>
<button (click)="showImage =! showImage" type="button"
    [class.przycisk]="enabled">Pokaż Zdjęcie</button>
```

[class.image]="enabled" – dodanie klasy **image** do **img**, oraz aktywowanie arkusza stylów celem zmiany wyglądu.

enabled – zmienna typu **boolean** znajdująca się w pliku: **app.component.ts**

```
<p>Użycie IF oraz ELSE:</p>
<p>Imię: {{osoba.name}}</p>
<p>Nazwisko: {{osoba.lastName}}</p>
 <br>
<button (click)="showImage =! showImage" type="button"
  [class.przycisk]="enabled">Pokaż Zdjęcie</button>
<ng-template #noImage>
  
</ng-template>
```

Powyżej użycie ng-template – brak obrazka profilowego

Dyrektywa ngFor – powtórzenie części kodu HTML na podstawie danych z tablicy

```
<p>ngFor</p>
<p>Imię: {{osoba.name}}</p>
<p>Nazwisko: {{osoba.lastName}}</p>
<ul>
  <li *ngFor="let value of osoba.adress">{{value}}</li>
</ul>
 <br>
<button (click)="showImage =! showImage" type="button"
  [class.przycisk]="enabled">Pokaż Zdjęcie</button>
```

Co nowego pojawiło się?

Otóż obiekt **osoba** musi posiadać tablicę której poszczególne elementy zawierają kolejne składowe adresu. (uwaga zmień interfejs Osoba).

Wynik w przeglądarce internetowej → → → → →

ngFor

Imię: Jan

Nazwisko: Brzechwa

- Kościuszki
- 4
- Nowa Sól
- 67-100

Pokaż Zdjęcie

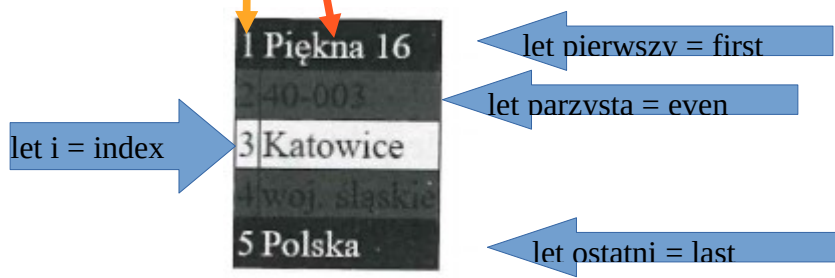
Dodatkowo można użyć zmiennych dla dyrektywy ngFor:

Nazwa zmiennej	Opis
index	Wartość typu number określająca położenie bieżącego obiektu
odd	Wartość logiczna; true oznacza, że położenie bieżącego obiektu jest określone liczbą nieparzystą
even	Wartość logiczna; true oznacza, że położenie bieżącego obiektu jest określone liczbą parzystą
first	Wartość logiczna; true oznacza, że bieżący obiekt jest pierwszym
last	Wartość logiczna; true oznacza, że bieżący obiekt jest ostatnim

Użycie w kodzie HTML:

```
<div id="kontakt" class="tekst">
  <table>
    <tr *ngFor="let item of osoba.adres; let i = index; let parzysta =
even; let pierwszy = first; let ostatni = last"
      [class.parzysta]="parzysta" [class.nieparzysta]="!parzysta" [class.
wyznienienie]="pierwszy || ostatni">
      <td>{{i+1}}</td>
      <td>{{item}}</td>
    </tr>
  </table>
</div>
```

Zwróć uwagę że do zmiany wyglądu używamy klas dodanych do naszego elementu!!! Wynik na stronie:



Dyrektywa ngSwitch – wybór odpowiedniej części kodu HTML do wyświetlenia

Teraz musimy dodać enum dla płci. Tworzymy nowy plik `Plec.ts` w folderze `interfejsy` (można dodać gdziekolwiek w projekcie ale najlepiej mieć wszystko ładnie posegregowane)

```
export enum Plec{
    k = 1,
    m,
    und
}
```

Teraz należy do naszego interfejsu – Osoby dodać płęć:

```
import { Plec } from './Plec';

export interface Osoba{
    name:String;
    lastName: String;
    picture:String;
    adress:String[];
    sex:Plec;
```

Zauważ że na górze
pliku musi być
import!!

Teraz należy podobnie dopisać do pliku `app.component.ts` import `Plec`. Dodatkowo należy dopisać płęć do obiektu osoby także w pliku `app.component.ts`:

```
osoba:Osoba = {
    name: "Jan",
    lastName: "Brzechwa",
    picture: "assets/images/osoba1.jpg", //folder assets/images/
    adress:["Kościuszki", "4", "Nowa Sól", "67-100"],
    sex: Plec.m
}
```

Teraz możemy wyświetlić odpowiedni obrazek w zależności od płci.

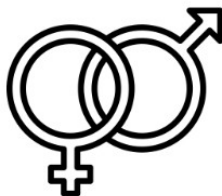
```
<p>ngSwitch</p>
<p>Imię: {{osoba.name}}</p>
<p>Nazwisko: {{osoba.lastName}}</p>
<ng-container [ngSwitch]="osoba.sex">
    
    
    
</ng-container>
<ul>
    <li *ngFor="let value of osoba.adress">{{value}}</li>
</ul>
 <br>
<button (click)="showImage = ! showImage" type="button"
    [class.przycisk]="enabled">Pokaż Zdjęcie</button>
```

W zależności jaką płeć ustawimy w pliku **app.component.ts** podczas tworzenia obiektu **osoba** taki otrzymamy wynik (wygląd strony internetowej): u mnie płeć: **und**.

ngSwitch

Imię: Jan

Nazwisko: Brzechwa



- Kościuszki
- 4
- Nowa Sól
- 67-100

DYREKTYWY ATRYBUTOWE: zmiana wyglądu i zachowania elementów drzewa dokumentów

Dyrektywa **ngStyle** – czyli formatowanie elementów

Dodaj do pliku **app.component.ts**:

```
kolory:string[] = ["blue","yellow","green","red"];  
wybor:number = 0;
```

```
//funkcja - prosta zmiana kolorów
```

```
zmiana():void{  
  this.wybor++;  
  if(this.wybor>=4) this.wybor = 0;  
}
```

Teraz w naszym kodzie HTML wpisujemy jak poniżej:

```
<button type="button" (click) = "zmiana()"> Zmień styl</button>  
<h1 [ngStyle] = "{ 'font.size.%':wybor*20+100,  
  'color':'white','background-color':kolory[wybor]} ">  
NAGŁÓWEK  
</h1>
```

Przetestuj działanie!!!

Dyrektywa ngClass – ustawienie klasy elementu

Masz następujące zdefiniowane klasy stylów (css)

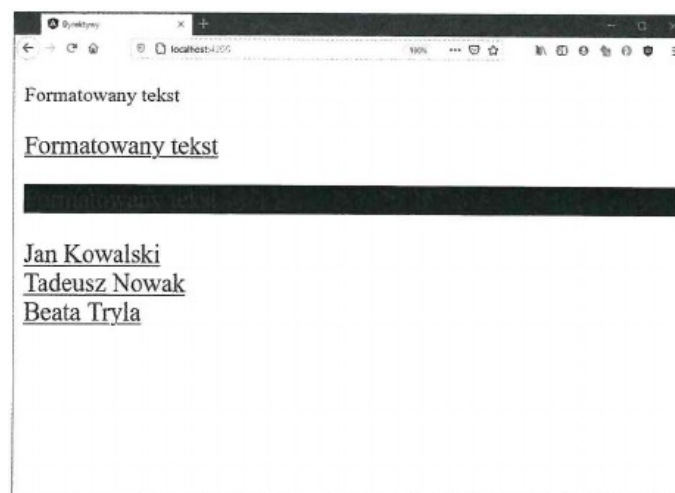
```
.a {  
    color: blue;  
}  
.b {  
    font-size: 20px;  
    text-decoration: underline;  
}  
.c {  
    background-color: black;  
}
```

Przypisujesz je do kodu HTML:

```
<p [ngClass]='a'>Formatowany tekst</p>  
<p [ngClass]='a b'>Formatowany tekst</p>  
<p [ngClass]='a b c'>Formatowany tekst</p>
```

```
<div *ngFor="let osoba of osoby" [ngClass]='a b'>{{osoba}}</div>
```

Wynik działania:



3adania

Stwórz nowy projekt Angula o nazwie DYREKTYWY_NAUKA. Pamiętaj aby użyć dyrektyw trzeba dopisać kilka linijek kodu do wygenerowanych plików →

14_angular_wiazanie_dwukierunkowe.pdf.

Zrób stronę podobną do tej w przykładach wyżej. - NA OCENĘ z wagą 1.

Druga ocena: waga 5

1. Przygotuj krótki opis miejscowości w której mieszkasz, oraz dołącz do niego dwa zdjęcia. Opis i zdjęcia wykorzystaj do stworzenia szablonu. Dodatkowo pod opisem umieść dwa przyciski i z użyciem dyrektywy **ngIf** spraw aby kliknięcie każdego przycisku spowodowało wyświetlenie jednego zdjęcia.
2. Zdefiniuj obiekt **kino** (jako interface), następnie na jego podstawie stwórz obiekt, w którym umieścisz następujące dane: nazwa, zdjęcie i adres **w tablicy** w formacie: [ulica, kod pocztowy, miejscowość]. Wyświetl dane zapisane w stworzonym obiekcie, jednak do wyświetlenia adresu użyj dyrektywy **ngFor** tak aby adres wyświetlił się w formie tabeli z obramowaniem.
3. Stwórz enum → **RodzajFilmu** w którym umieść wyliczenia: **sensacyjny=1, kryminał, nieokreślono**. Następnie dodaj do obiektu **kino** (zad 2) pole **rodzajFilmu** typu **RodzajFilmu**; Następnie stwórz dwa kina z filmem sensacyjnym i kryminałem. Następnie przygotuj trzy obrazki symbolizujące: film sensacyjny, kryminał i nieznany. Wyświetl dane kin na stronie HTML oraz odpowiedni obrazek za pomocą dyrektywy **ngSwitch**.
4. Do strony dodaj możliwość zmiany koloru czcionki (do wyboru: czerwony, niebieski i zielony) po naciśnięciu przycisku – zmień kolor czcionki w sposób ciągły to znaczy kolory będą się zmieniać po kolei: czerwony, niebieski, zielony i następnie znowu czerwony.