**Фреймворк Angular 9**

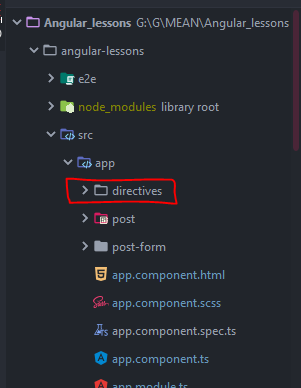
**Лабораторная работа № 3**

**Работа с директивами**

**Задача. Создать директиву для обработки события. Событие должно срабатывать при наведении мыши на пост и сбрасываться при переходе на другой пост.**

**Ход работы**

1. Создадим директорию в которой будут храниться диретивы.



1. В командной строке (терминале) введите:

**ng g d directives/ipz --skipTests**

В папке directives будет создан файл ipz.directive.ts со следующим содержимым

import { *Directive* } from '@angular/core';  
  
@Directive({  
 selector: '[appIpz]'  
})  
export class IpzDirective {  
  
 constructor() { }  
  
}

1. Для добавления события в директиве используют специальный декоратор @HostListener()

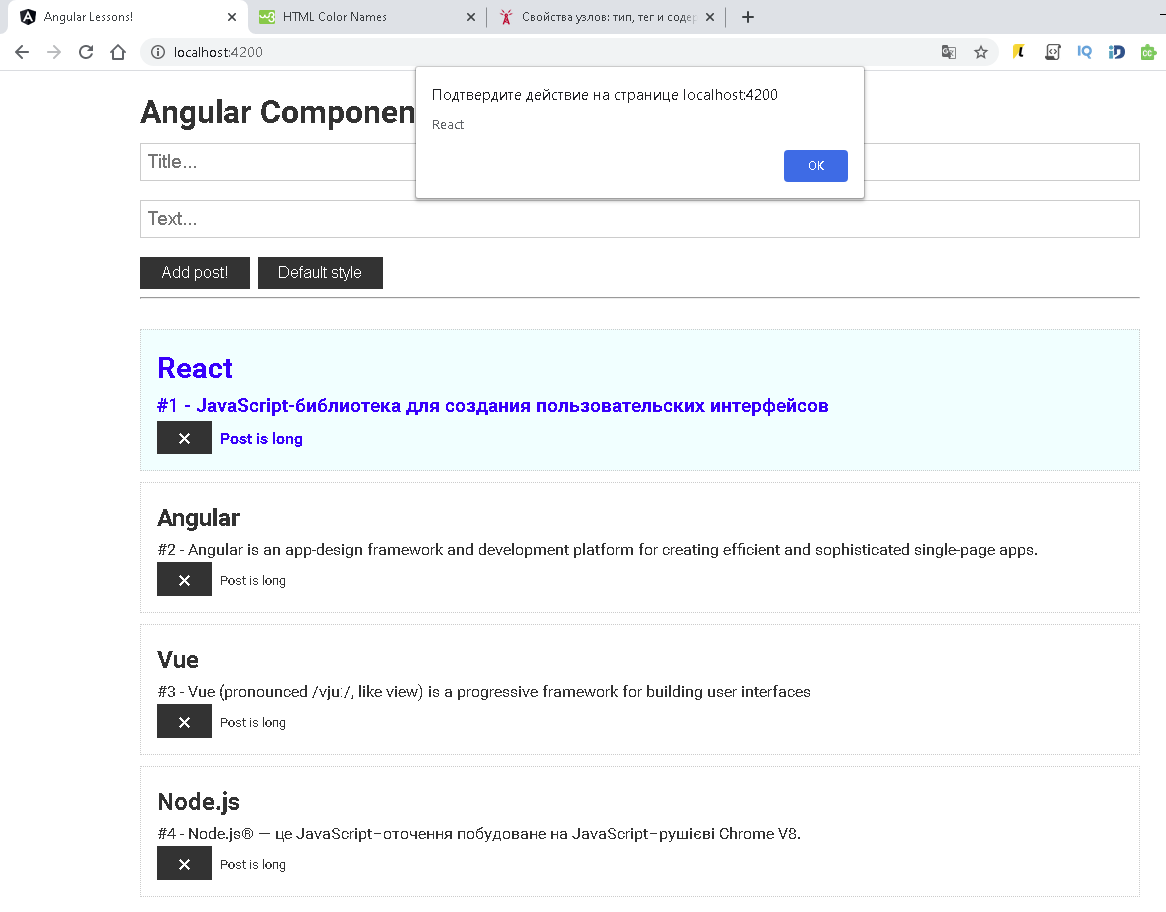
Добавим в класс в файле ipz.directive.ts код для обработки события наведения мышки на пост.

import {*Directive*, ElementRef, *HostListener*, Renderer2} from '@angular/core';  
  
@Directive({  
 selector: '[appIpz]'  
})  
export class IpzDirective {  
  
 constructor(private el: ElementRef, private r: Renderer2) {  
 }  
 @HostListener('mouseenter', ['$event.target']) onMouseEnter() {  
this.r.setStyle(this.el.nativeElement,'color','blue')  
this.r.setStyle(this.el.nativeElement,'margin-top','30px')  
this.r.setStyle(this.el.nativeElement,'font-weight','bold')  
this.r.setStyle(this.el.nativeElement,'font-size','19px')  
this.r.setStyle(this.el.nativeElement,'background-color','azure')  
  
 }  
 @HostListener('mouseleave', ['$event.target']) onMouseLive() {  
 this.r.setStyle(this.el.nativeElement,'color',null)  
 this.r.setStyle(this.el.nativeElement,'margin-top',null)  
 this.r.setStyle(this.el.nativeElement,'font-weight',null)  
 this.r.setStyle(this.el.nativeElement,'font-size',null)  
 this.r.setStyle(this.el.nativeElement,'background-color',null)  
  
 }  
  
}

1. Теперь в компоненте можем использовать созданный декоратор для любого html элемента. Добавим его в файл post.component.html

<div appIpz class**="card"**>  
 <h2>{{post.title}}</h2>  
 <p > **#**{{post.id}} **-** {{post.text}} </p>  
 <button class**="btn"** (click)**="**removePost()**"**>**&#10006;**</button>  
 <ng-content> </ng-content>  
</div>

**!!!Самостоятельно.**  **Cоздать декоратор обрабатывающий событие нажатия мышки на пост. В модельное окно (alert) должно выводиться название поста:**



1. Модифицировать директиву и сделать так чтобы можно было в нее передавать параметры. Для этого в шаблоне запишем следующим образом

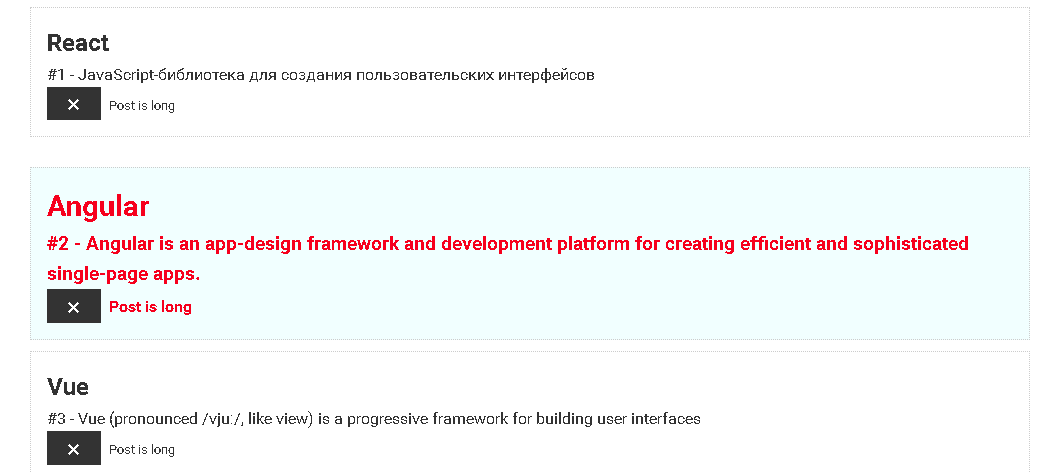
<div appIpz**="red"** class**="card"**>  
 <h2 >{{post.title}}</h2>  
 <p > **#**{{post.id}} **-** {{post.text}} </p>  
 <button class**="btn"** (click)**="**removePost()**"**>**&#10006;**</button>  
 <ng-content> </ng-content>  
</div>

А в файле ipz.directive.ts добавим декоратор @Input()

@Input('appIpz') color: string

И переменную color можно будет использовать в строчке

this.r.setStyle(this.el.nativeElement,'color', this.color)



Второй способ передачи параметров в директиву. В шаблоне post.component.thml заводим переменную например dStyle и передаем в нее объект:

<div appIpz**="red"** [dStyle] **= "**{marginTop: '30px' }**"** class**="card"**>  
 <h2 >{{post.title}}</h2>  
 <p > **#**{{post.id}} **-** {{post.text}} </p>  
 <button class**="btn"** (click)**="**removePost()**"**>**&#10006;**</button>  
 <ng-content> </ng-content>  
</div>

А в файле ipz.directive.ts принимаем параметры с помощью того же декоратор @Input()

@Input() dStyle: {marginTop?: string}

И меняем на динамику изменение верхней границы.

this.r.setStyle(this.el.nativeElement,'margin-top',this.dStyle.marginTop)

**!!!Самостоятельно.**  **Cоздать декоратор обрабатывающий событие нажатия мышки на пост. В модельное окно (alert) должно выводиться название поста:**