Task – 1 (on click button only odd numbers from array should be displayed)

.html File

<br>

<button class="btn btn-primary" (click)="flag=!flag">Only show odd numbers</button>

<br><br>

<div \*ngIf="flag==true">

<ul class="list-group"

\*ngFor="let i of numbers">

<li class="list-group-item"

\*ngIf="i%2==0">{{i}}</li>

</ul>

</div>

.ts File

export class AppComponent {

numbers=[1,2,3,4,5];

flag=false;

}

Task – 2 (Button as a toggle)

.html

<br>

<button class="btn btn-primary" (click)="onlyodd=!onlyodd">Only show odd numbers</button>

<br><br>

<ul class="list-group">

<div \*ngIf="onlyodd">

<li class="list-group-item"

\*ngFor="let odd of oddnum">{{odd}}</li>

</div>

<div \*ngIf="!onlyodd">

<li class="list-group-item"

\*ngFor="let even of evnnum">{{even}}</li>

</div>

</ul>

.ts

oddnum=[1,3,5];

evnnum=[2,4,6];

onlyodd=false;

Task – 3 (ngClass & ngStyle)

.css

.odd{

color:red;

}

.html

<br>

<button class="btn btn-primary" (click)="onlyodd=!onlyodd">Only show odd numbers</button>

<br><br>

<ul class="list-group">

<div \*ngIf="onlyodd">

<li class="list-group-item"

[ngClass]="{odd:odd%2!==0}"

[ngStyle]="{backgroundColor:odd%2!==0? 'yellow' : 'transparent'}"

\*ngFor="let odd of oddnum">{{odd}}</li>

</div>

<div \*ngIf="!onlyodd">

<li class="list-group-item"

[ngClass]="{odd:even%2!==0}"

[ngStyle]="{backgroundColor:even%2!==0? 'yellow' : 'transparent'}"

\*ngFor="let even of evnnum">{{even}}</li>

</div>

</ul>

.ts

export class AppComponent {

oddnum=[1,3,5];

evnnum=[2,4,6];

onlyodd=false;

}

Task – 4 (Attribute Directive with HostListener)

.ts (new for directive)

import { Directive, Renderer2, OnInit, ElementRef, HostListener } from '@angular/core';

import { NgModuleResolver } from '@angular/compiler';

@Directive({

selector: '[appHighlight]'

})

export class HighlightDirective implements OnInit{

constructor(private elRef:ElementRef, private renderer: Renderer2) { }

ngOnInit()

{

//this.renderer.setStyle(this.elRef.nativeElement,'background-color','blue');

}

@HostListener('mouseenter') mouseover(eventData: Event)

{

this.renderer.setStyle(this.elRef.nativeElement,'background-color','blue');

}

@HostListener('mouseleave') mouseleave(eventData: Event)

{

this.renderer.setStyle(this.elRef.nativeElement,'background-color','transparent');

}

}

.ts

.html

<p appHighlight>This is HostListener Demo</p>

Task-5 (HostBinding)

Change to only .ts file in directive

Add in class as first:

@HostBinding('style.backgroundColor') backgroundColor: string='transparent';

@HostListener('mouseenter') mouseover(eventData: Event)

{

// this.renderer.setStyle(this.elRef.nativeElement,'background-color','blue');

**this.backgroundColor='blue';**

}

@HostListener('mouseleave') mouseleave(eventData: Event)

{

// this.renderer.setStyle(this.elRef.nativeElement,'background-color','transparent');

**this.backgroundColor='transparent';**

}