**Angular Components**

**Dropdown**

**HTML**

<select multiple placeholder="Select"(change)="onChange($event)" [disabled]="isDisabled" (blur)="onBlur($event)" [ngModel]="selectedDeviceObj" (ngModelChange)="onChangeObj($event)" name="name">

<option \*ngFor="let v of values" [value]="v.id">{{v.name}}</option>

</select>  
  
**TS**selectedDeviceObj:any;  
values:any = [‘a’,’b’,’c’]onChange($event) {}onBlur($event) {}

**Toggle**

**HTML**

<mat-slide-toggle [checked] = "checked" [disabled] = "disabled">Slide!</mat-slide-toggle>

**TS**checked:Boolean = false;  
disabled:Boolean = true;

**Slider**

**HTML**

<mat-slider class="example-margin"

[disabled]="disabled"

[invert]="invert"

[max]="max"

[min]="min"

[step]="step"

[thumbLabel]="thumbLabel"

[tickInterval]="getSliderTickInterval()"

[(ngModel)]="value"

[vertical]="vertical"

aria-labelledby="example-name-label">

</mat-slider>

**TS**

autoTicks = false;

disabled = false;

invert = false;

max = 100;

min = 0;

showTicks = false;

step = 1;

thumbLabel = false;

value = 0;

vertical = false;

tickInterval = 1;

|  |  |  |
| --- | --- | --- |
| Button Tag | Html Code | Ts code |
| “onClick” | (click)="myClickFunction($event)" | myClickFunction(event: Event) { console.log(‘Click!’ , event) |
| “clickBinding” | <button (click)="myFunction()"></button> | myFunction(event: Event) { console.log(event) |
| “disableClick” | <button [disabled]="isDisabled"(click)="btnClick.emit($event)"> | onButtonClick(event :Event){  (event.target as HTMLButtonElement).disabled = true; |