Attribute Directives: - This type of directive changes behaviors of DOM element.

3 Different Way to create Form ()

1. Template Driven Form (<https://www.youtube.com/watch?v=_hso8Z2zRnI&index=17&list=PLC3y8-rFHvwg5gEu2KF4sbGvpUqMRSBSW>)
2. Model Driven Form
3. Model Driven From- Form Builder

* Generate Component

ng g c test

or

ng generate component [name]

Class Binding:-

.class-danger{

color: red;

}

.class-sucess{

color: green;

}

<p [class]="classDanger">Test para1</p>

<p [class.class-sucess]="IsSuccess">Test para2</p>

<p [class]="clasSucess">Test para3</p>

<p [ngClass]="messageClass">Test para4</p>

public classDanger="class-danger";

public clasSucess="class-sucess";

public IsSuccess=true;

public messageClass={

"class-sucess":this.IsSuccess,

"class-danger":!this.IsSuccess

}

Style Binding

<h4 [style.color]="IsSuccess?'green':'red'">Style binding</h4>

<h4 [class]="clasSucess">Style binding 2</h4>

<h4 [ngStyle]="titleStyle">style binding 3</h4>

public clasSucess="class-sucess";

public IsSuccess=false;

public titleStyle={

color:"green",

fontStyle:"italic"

}

Two Way Binding

<input type="text" [(ngModel)]="Name">

{{Name}}

public Name ="";

**app.module.ts**

import { FormsModule } from '@angular/forms';

imports: [

BrowserModule,

AppRoutingModule,FormsModule

],

Template Reference variables:

When there is user interaction, we might want some data flow from user view to class to perform an operation

<input #myName type="text">

<button (click)="LogMessage(myName.value)">click Here</button>

<br>

{{Name}}

public Name ="";

public LogMessage(value){

this.Name=value;

}

Structural Directives:

Structural directives are directives that let you add or remove HTML elements from the DOM. Following are three built-in directives-

* Ngif
* ngSwitch
* ngFor

First tow directives are use to conditionally render the html elements and ngFor directive to use to render list of html elements.

ngIf

1.

<h4 \*ngIf="true">Hello India</h4>

<span>following will not render in DOM</span>

<h4 \*ngIf="false">Hello India</h4>

2.

<h4 \*ngIf=displayElement>Hello India</h4>

public displayElement =false;

3.

<h4 \*ngIf= "displayElement; else elseBlock">Hello India</h4>

<ng-template #elseBlock>

<h3>Name is hidden</h3>

</ng-template>

public displayElement =true;

4.

<div \*ngIf="displayElement; then thenBlock; else elseBlock"></div>

<ng-template #elseBlock>

<h3>Name is hidden</h3>

</ng-template>

<ng-template #thenBlock>

<h3>Name is display</h3>

</ng-template>

public displayElement =true;

ngSwitch

<div [ngSwitch]="color">

<div \*ngSwitchCase="'Green'">Color Green</div>

<div \*ngSwitchCase="'Red'">Color Red</div>

<div \*ngSwitchCase="'Blue'">Color Blue</div>

<div \*ngSwitchDefault>Color Yellow</div>

</div>

public color ="Blue";

ngFor

public colors =["Red","Green","Blue","Yellow"];

<div \*ngFor="let color of colors">

{{color}}

</div>

Output:-

Red

Green

Blue

Yellow

<div \*ngFor="let color of colors; index as i">

{{i}} {{color}}

</div>

Output:-

0 Red

1 Green

2 Blue

3 Yellow

To know even and odd element in index we can do like this

<div \*ngFor="let color of colors; even as e">

{{color}}:- {{e}}

</div>

**Output:-**

Red: - true

Green: - false

Blue: - true

Yellow: - false

Even as e for even

Odd as o for odd

Last as l for last element in loop

First as f for fist element in loop