Child.component.ts

import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css']

})

export class ChildComponent {

  obj = { cname: "Shrilata-child-data",

          caddress: "Aundh-child-data" }

  @Input('parentData')

  public strmsg;

  @Output()

  public myevent = new EventEmitter();

  onclick() {

    this.myevent.emit(this.obj);

  }

}

Child.component.html

<div>In Child :->  Data from Parent :  {{strmsg}}</div>

<button (click)=onclick()>click me</button>

parent.component.ts

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-parent',

  templateUrl: './parent.component.html',

  styleUrls: ['./parent.component.css']

})

export class ParentComponent {

  pData = "data-from-parent";

  public message;

}

parent.component.html

In Parent :-> Data From Child Component : {{message.cname}}

<app-child [parentData]="pData" (myevent)="message=$event"></app-child>

Model.ts

export class Model {

  user;

  items;

  contact;

  constructor() {

    this.user = "Shrilata";

    this.contact = {

      email: "shrilata@gmail.com",

      phone: 9977886600

    };

    this.items = [new TodoItem("Buy Flowers", false),

                  new TodoItem("Get laundry", false),

                  new TodoItem("Collect Tickets", false),

                  new TodoItem("Call Mom", false)]

  }

}

export class TodoItem {

  action;

  done;

  constructor(action, done) {

    this.action = action;

    this.done = done;

  }

}

User-model.component.ts

import { Component, OnInit } from '@angular/core';

import { Model } from './model';

@Component({

  selector: 'app-user-model',

  templateUrl: './user-model.component.html',

  styleUrls: ['./user-model.component.css']

})

export class UserModelComponent {

  model = new Model();

  getName(){

    return this.model.user;

  }

  getToDoItems(){

    return this.model.items;

  }

  constructor() { }

}

User-model.component.html

<h2>{{getName()}}'s ToDo List</h2>

<table border="1" style="border-collapse:collapsed">

    <tr><th>Sr no</th><th>Description</th></tr>

    <tr \*ngFor="let item of getToDoItems(); let i = index">

       <td>{{ i + 1 }}</td>

       <td>{{ item.action }}</td>

    </tr>

 </table>

User-manage.component.ts

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-user-manage',

templateUrl: './user-manage.component.html',

styleUrls: ['./user-manage.component.css']

})

export class UserManageComponent implements OnInit{

titlelabel:string="Cricket Players";

names = ["Virat","Yuvraj","Dhoni"];

addName(name){

this.names.push(name);

}

delName(name){

for(var i in this.names){

if(this.names[i]==name){

this.names.splice(parseInt(i),1);

console.log("found"); break;

}

}

}

delAllNames(){

this.names.splice(0,this.names.length);

}

showEvent(event){

console.log(event.target.value);

}

hide(event){

event.target.style.visibility = 'hidden';

}

show(event){

event.target.style.visibility = 'visible';

}

constructor() { }

ngOnInit() { }

}

User-manage.component.html

<fieldset>

<br>

    <h2>{{titlelabel}}</h2>

<h4>

  <ul><li \*ngFor="let n of names">{{n}}</li></ul>

</h4>

Player Name:<input name="t1" #player><br>

<input type="submit" value="add player" (click)="addName(player.value)">

<input type="submit" value="del player" (click)="delName(player.value)">

<input type="submit" value="del players" (click)="delAllNames()">

</fieldset>

<br><br>

Enter name to see value on console: <input type="text" value="Shrilata T" (click)="showEvent($event)">

<br>

Enter value in text to see value : <input type="text" (keyup)="showEvent($event)">

<br>

Click on button to see target : <input type="button" value="A button"(click)="showEvent($event)">

<br>

<h2 style="color:red">Move mouse over "Div Element" to show/hide it </h2>

<h2><div (mouseover)="hide($event)"

         (mouseout)="show($event)">Div element</div></h2>

User.ts

export class User {

constructor(public name: string,

public age: number) {

}

}

Structural-directive.component.ts

import { Component, OnInit } from '@angular/core';

import {User} from './user';

@Component({

selector: 'app-structural-directive',

templateUrl: './structural-directive.component.html',

styleUrls: ['./structural-directive.component.css']

})

export class StructuralDirectiveComponent {

users = [

new User('Mahesh', 20),

new User('Krishna', 22),

new User('Narendra', 30)

];

isValid = true;

ids = [1,2,3,4];

user1 = new User("Shrilata",20);

user2:User;

data2=0;

}

Structural-directive.component.html

<b> \*ngFor Demo</b>

<ul>

<li \*ngFor="let user of users">

{{user.name}} - {{user.age}}

</li>

</ul>

<b>index variable demo </b>

<p \*ngFor="let user of users; let i = index">

Row {{i}} : Name: {{user.name}}

</p>

<b>first and last variable demo </b>

<div \*ngFor="let user of users; let i = index; let f=first; let l=last;">

Row {{i}} : Name: {{user.name}}, is first row: {{f}}, is last row: {{l}}

</div>

<b>even and odd variable demo </b>

<div \*ngFor="let user of users; let i = index; let e=even; let o=odd;">

Row {{i}} : Name: {{user.name}}, is even row: {{e}}, is odd row: {{o}}

</div>

<!--

<br/><b>ngFor with component element using \*ngFor</b>

<emp-app \*ngFor="let user of users" [emp]="user"> </emp-app>

<br/><b>ngFor with component element using template directive </b>

<emp-app template="ngFor let user of users" [emp]="user"> </emp-app>

<br/><b>ngFor with component element using template tag </b>

<template ngFor let-user [ngForOf]="users">

<emp-app [emp]="user"></emp-app>

</template>

-->

<b>NgIf with HTML Elements </b><br/>

<p \*ngIf="isValid">

Data is valid.

</p>

<p \*ngIf="!isValid">

Data is not valid.

</p>

<div \*ngFor="let id of ids">

Id is {{id}}

<div \*ngIf="id%2 == 0">

<div [ngClass]="'one'">Even Number</div>

</div>

<div \*ngIf="id%2 == 1">

<div [ngClass]="'two'">Odd Number</div>

</div>

</div>

<br/>

<div \*ngIf="user1">

Age:{{user1.age}} Name: {{user1.name}}

</div>

<div \*ngIf="user2">

Age:{{user2.age}} Name: {{user2.name}}

</div>

<!--<br/><b>NgIf with Component Elements</b><br/>

<my-msg \*ngIf="emp1" [pname] = "emp1.name"> </my-msg>

<my-msg template="ngIf:emp1" [pname] = "emp1.name"> </my-msg>

<template [ngIf]="emp1">

<my-msg \*ngIf="emp1" [pname] = "emp1.name"> </my-msg>

</template>

-->

Input using ngModel : <input [(ngModel)] ='data2'> <br>

<div [ngSwitch]="data2">

<p \*ngSwitchCase="1">You selected One</p>

<p \*ngSwitchCase="2">You selected Two</p>

<p \*ngSwitchCase="3">You selected Three</p>

<p \*ngSwitchDefault>Sorry Invalid selection!!</p>

</div>