

## Angular: ❤️ Day3

- In angular can do for loop in html part
- In app Module add in Imports part **FormsModule** to twoway binding work and import this  
**import {FormsModule} from '@angular/forms'**
- While creating component by command and need to delete testing files use this command  
**ng g c compName - -skip-tests**

### ➤ Built in directive

#### 1. \*ngFor "Structural directive" (take for loop 3adiya)

```
<tr *ngFor="let s of Students">
  <td>{{s.name}}</td>
  <td>{{s.age}}</td>
</tr>
```

#### 2. \*ngIf "take condition "

Note -> Ecma Script syntax to convert something to integer add before it +

```
<tr *ngFor="let s of Students">
  <td>{{s.name}}</td>
  <td *ngIf="+s.age<50">{{s.age}}</td>
</tr>
```

#### 3. If else "by ng-template"

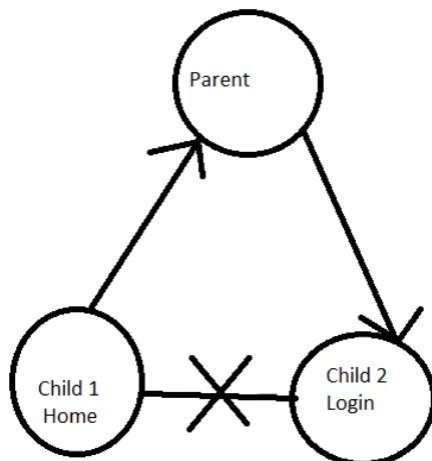
```
<div *ngIf="+age>50 else hamada";>
  <p style="color: red;">Age Not Valid</p>
</div>
<ng-template #hamada >//reference
  <p style="color: green;">Age Valid</p>
</ng-template>
```

## 4. Nested if

```
<div *ngIf="+age>50 else hamada">
  <p style="color: red;">Age Not Valid</p>
</div>
<ng-template #hamada >
  //nested in p tag
  <p *ngIf="age" style="color: green;">Age Valid</p>
</ng-template>
```

➤ Component Interaction "send data from component to another"

Send data first to parent then parent send it to another component



### 1. Parent to child

Steps: to get data from parent to home component

1) Make property in app-home and path to it parent property

```
<div>
  <app-home [hamadaa]="ParentData"></app-home>
</div>
```

2) In home.ts declare hamadaa property

```
export class HomeComponent {
  hamadaa="";
}
```

### 3) Must declare input decorator in home

```
export class HomeComponent {  
  @Input() hamadaa="";  
}  
//to work add Input in import
```

### 4) Display it in html

## 2. Child to parent

Steps: need to send data from login component to parent

1) Add property to login.ts

2) Need event to path data to parent

2.1) add it in app-login

```
<div>  
  <app-login (myEvent)="getData($event)"></app-login>  
</div>
```

getdata(data:any)//in app.ts

myEvent//in login.ts and datatype must be

@output

```
export class LoginComponent {  
  @Output() myEvent=new EventEmitter();  
}
```

## 3. Child to child "through parent"

Need to send data from login to home

```
<div>  
  <app-home [hamadaa]="datafromlogin"></app-home>  
</div> //الى على اليمين خاص بالهوم والشمال الفاليو
```

ينفع لل property نديها **Alise name** بس برا لازم تنادي بيه انما فى الكلاس بالاسم العادى

Ex:

```
export class HomeComponent {  
  @Input("dala3") hamadaa="";  
}
```

External Material:

Component lifecycle:

1. Ts file
2. constructor
3. Built in functions

اول فانكشن بتننفذ بعد ال constructor هيا (**ngonchange**) بيقرأ فيها اى input parameter جاية لل component  
تانى واحدة (**ngOninit**) لو محتاج اى حاجة بتننفذ بعد ما ال component يفتح بنعملها فى ال oninit  
التالتة (**ngDocheck**) ليها علاقة بالحاجات اللى بعملها ngfor

4. ngAfterContentInit()

5. ngAfterContentChecked()

لحد ما يوصل هنا بيكون لسه مرحش ال view ولا يعرف حاجة عنة

6. NgAfterViewInit()

هنا يبدأ يروح لـ view (لو عايزة اتعامل مع اى حاجة موجودة فى ال view مش هينفع الا لما يوصل لل function دى )

7. ngAfterviewChecked()

8. ngOnDestroy()

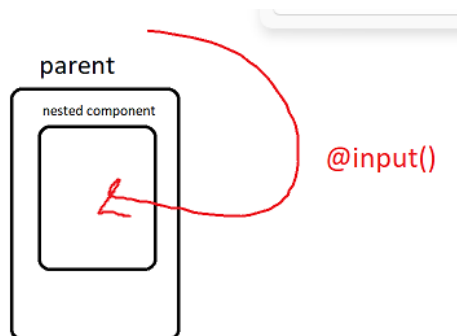
بتتنفذ لما ال component يحصله unload

## Note:

Angular شغالة dependency injection → ای حابه عایز تعملها initialization  
حطها فی ال constructor

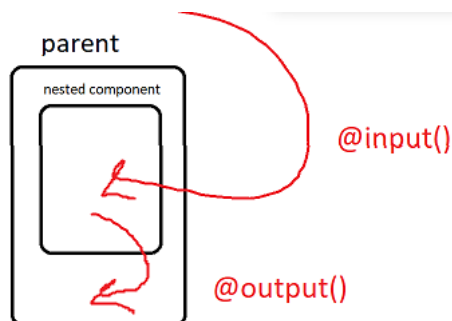
## Component Interaction : "Sharing data"

### 1) parent to nested component



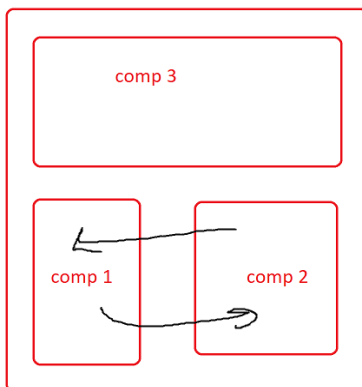
هنا ال nested عایز یقرا حاجة موجودة عند آل parent == هنسمیه input جای من آل  
parent ل nested إذن هنستخدم معاه ال input decorator function

### 2) nested component to parent



هنا العکس بقى من ال nested ل parent یعنی عایز اطلع داتا إذن استخدم output  
decorator function

### 3)Between 2 component doesn't have relation



هنا ال input وال output مش هتشتغل لانهم مش **nested**  
الحل هيكون بال **services[Advanced]**