**26-04-2022**

**We can create the angular component using ng command**

**Syntax**

**ng generate component componentname**

**or**

**ng g c componentname**

**ng new angular-data-binding**

**Angular Data binding : it is use to share the data between component(ts) and view (HTML or template).**

1. **One way data binding**
2. **String interpolation : Component to View**

**Component ----------------------------🡪View**

**{{}}**

**{{variableName}}**

**{{expression}}**

**{{display()}}**

**{{sayHello(“Ravi”)}}**

**ng g c string-interpolation**

1. **Property binding : Component to View**

**Component -------------------------------🡪View**

**[]**

**Component contains**

**lname:string =”patil”**

**in html page**

**<input type=”text” value=”lname”/> pure html page**

**<input type=”text” [value]=”lname”/> angular will check lname variable inside a component if present it will display the value of lname variable inside a text field.**

**<p [innerText]=”lname”> </p>**

**<div [innerText]=”lname”> </div>**

**ng g c property-binding**

**{{}} : we display the value of variable using string interpolation without tag.**

**In String interpolation always value is consider as string.**

**[] : we have take the help dom (html tag) to achieve property binding.**

**But in property binding we can use different type of data types.**

1. **Event binding : View ----------------------🡪 Component**

**Angular use same event provided by JavaScript only difference they remove pre-fix on and all event name wrap with ().**

**JS Event Angular Event**

**onClick (click)**

**onDblClick (dblclick)**

**onChange (change)**

**onSubmit (ngSubmit)**

**etc etc**

**JS Event is use to call JavaScript function.**

**Angular event is use to call Typescript function part of component and we are calling those function without creating the object of that class.**

**ng g c event-binding**

**event binding with string interpolation or property to achieve two binding.**

**Passing the value from view to component**

1. **Using template reference :**

**<input type=”text” #nameRef />**

1. **Two way data binding**