# Ceate Angular Template

## 1. Setup Project

### Install Dependencies

1. Change directory to calab:

* cd calab

1. Install dependencies by running the following command:

* npm install

## 2. Create Component and Temlate

### 2.1 Create a new Component and a Template

1. Create a new CustomInputComponent component using CLI:

* npx -p @angular/cli ng generate component custom-input

### 2.2 Inject CustomInput Component into AppComponent

1. Inside src/app/app.component.ts do the following:
   * Import CustomInputComponent:

* import { CustomInputComponent } from './custom-input/custom-input.component';
  + Update imports to include CustomInputComponent:
* imports: [RouterOutlet, CustomInputComponent],

1. Open src/app/app.component.html template and add the following element after the <div class="divider"...

* <app-custom-input></app-custom-input>

### 2.3 Start The Application

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve
* *Otherwise refresh the browser tab to see updated view.*

## 3. Data Binding Examples

### 3.1 Text Interpolation Example

1. Open src/app/custom-input/custom-input.component.html and replace current html content with the following:

* <p>{{value}}</p>

1. Open src/app/custom-input/custom-input.component.ts and add the following code:

* export class CustomInputComponent {  
   value: string = "My Default Value"  
  }
* *Open your Angular application in Browser and see the result.*

### 3.2 Event Binding Example

1. Open src/app/custom-input/custom-input.component.html and add the following line just above <p>:

* <input type="text" (input)="onInputChange($event)">

1. Open src/app/custom-input/custom-input.component.ts and add the following code inside CustomInputComponent class:

* onInputChange(event: any) {  
   this.value = event.target.value;  
  }
* *Open your Angular application in Browser and see the result. Enter any value into the TextInput box and see how value chages on a screen.*

### 3.3 Property Binding Example

1. Open src/app/custom-input/custom-input.component.html and update current HTML:
   * add value property to <input> element:
   * <input type="text" [value]="value" (input)="onInputChange($event)">

* *Open your Angular application in Browser (refresh page if needed) and see “My Default Value” being pre populated in TextInput box.*

### 3.4 Two Way Binding Example

1. Open src/app/custom-input/custom-input.component.html and remove current <p>{{value}}</p> from Template.:
2. Open src/app/custom-input/custom-input.component.ts and update to the following code:

* import { Component, EventEmitter, Input, Output } from '@angular/core';  
   @Component({  
   selector: 'app-custom-input',  
   standalone: true,  
   imports: [],  
   templateUrl: './custom-input.component.html',  
   styleUrl: './custom-input.component.css'  
   })  
   export class CustomInputComponent {  
   @Input() value: string | undefined;  
   @Output() valueChange = new EventEmitter<string>();  
    
   onInputChange(event: any) {  
   this.value = event.target.value;  
   this.valueChange.emit(this.value);  
   }  
   }

1. Open src/app/app.component.ts and just below title variable add the following:

* inputValue: string = 'initial value';

1. Open src/app/app.component.html template and update/add the following element below the <div class="divider"...

* <app-custom-input [(value)]="inputValue"></app-custom-input>  
  <p>Input value: {{ inputValue }}</p>
* *Open your Angular application in Browser (refresh page if needed) and see how default value is set initialy, but later updated when new value is typed into TextInput box.*

### 3.5 Start The Application

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve
* *Otherwise refresh the browser tab to see updated view.*