**Lesson02 Ng Model in custom Directive**

**Notes:-**

**1-Angular Ng Model Directive Example Tutorial is the today’s leading topic. The ng-model directive binds the value of HTML controls (input, select, and text area) to application data.**

**2-Using the two-way binding, we can display a data property as well as an update that property when the user makes changes. We can merely achieve it in the component element as well as HTML element both. The two-way binding uses the syntax as [()] or bind- keyword.**

**3-The two-way binding uses the syntax of property binding and event binding together. Property binding uses the syntax as bracket [] or bind- and event binding uses the syntax as the parenthesis () or on- and these bindings are considered as one-way binding. Two-way binding works in both directions are setting the value and fetching the value. The two-way binding uses a specific name pattern.**

**Steps:-**

**1-on the app.module.ts we import the**

**import { BrowserModule } from '@angular/platform-browser';**

**import { NgModule } from '@angular/core';**

**//we import the form module on the app.module.ts**

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

**import { AppComponent } from './app.component';**

**@NgModule({**

**declarations: [**

**AppComponent],**

**imports: [**

**BrowserModule,**

**//we import forms module**

**FormsModule],**

**providers: [],**

**bootstrap: [AppComponent]})**

**export class AppModule { }**

**2-on the angular project create custom classes**

**export class AppData {**

**constructor(public name: String) {}}**

**3-on the component .ts we add the following code**

**import { Component } from '@angular/core';**

**import { AppData } from './AppData';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']})**

**export class AppComponent {**

**title = 'app';**

**data = new AppData('');}**

**4-on the component.html we add the following code**

**<input type="text" class="form-control" id="name" required**

**[(ngModel)]="data.name" />**

**<p>Hello {{ data.name }}!</p>**