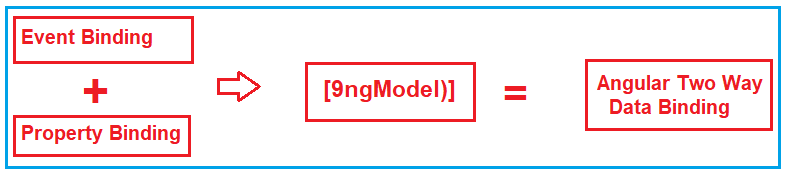
**What is Angular Two Way Binding?**

The most popular and widely used data binding mechanism in Angular Application is two-way data binding. The two-way data binding is basically used in the input type filed or any form element where the user type or provide any value or change any control value on the one side and on the other side, the same automatically updated into the component variables and vice-versa is also true.



The two-way data binding in Angular is actually a combination of [**Property Binding**](https://dotnettutorials.net/lesson/angular-property-binding/) and [**Event Binding**](https://dotnettutorials.net/lesson/angular-event-binding/). The Syntax is given below:

**<input [value] = ‘data 1’ (input) = ‘data = $event.target.value’>**

**Two-Way Binding using ngModel Directive:**

You can also implement the two-way data binding in Angular Application using the ngModel directive. The ngModel directive combines the square brackets of property binding with the parentheses of event binding in a single notation. The syntax to use ngModel for two-way data binding is given below.

**<input [(ngModel)] = ‘data’>**

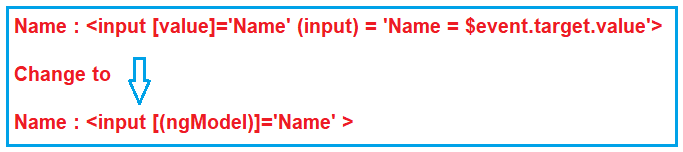
If you are getting confused at the moment, then don’t worry, we will try to explain two-way data binding using both the approaches.

**Example to understand Angular Two Way Data Binding:**

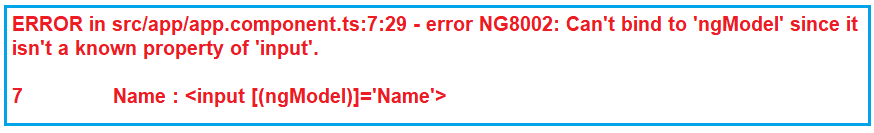
Let’s directly start with an example. First, modify the **app.component.ts** file as shown below.

**wo Way Data binding using ngModel Directive:**

To simplify the two-way data binding, the angular framework has provided one directive called the ngModel directive. With the ngModel directive, you can change to existing code as shown below.



With the above changes in place in the app.component.ts file, at this point, if you run the application, then you will get the following error.



This is because the **ngModel**directive is available in the system module called **FormsModule**. If you want to use the **ngModel**directive, then in your root module that is **AppModule**, you will have to import the **FormsModule**first.

**Steps to use ngModel Directive:**

Here are the steps to import FormsModule into our AppModule

1. Open **app.module.ts** file  
2. Include the following import statement in it  
       **import { FormsModule } from ‘@angular/forms’;**  
3. Also, include FormsModule in the ‘imports’ array of @NgModule  
       **imports: [BrowserModule, FormsModule]**

With the above changes in place, the complete code of **App.Module.ts** is as follows.

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

**import** **{** AppRoutingModule **}** from './app-routing.module';

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

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

@NgModule**({**

declarations: **[**

AppComponent

**]**,

imports: **[**

BrowserModule,

AppRoutingModule,

FormsModule

**]**,

providers: **[]**,

bootstrap: **[**AppComponent**]**

**})**

**export** **class** AppModule **{** **}**

**The complete codes of app.component.ts file as follows:**

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

@Component**({**

selector: 'app-root',

template: `<div>

Name : <input [(ngModel)]='Name'>

<br>

You entered : {{Name}}

</div>`

**})**

**export** **class** AppComponent **{**

Name: **string** = 'Geetha';

**}**

With the above changes, now reload the web page and you should see everything is working as expected.