Understanding Angular

Angular Introduction

Angular is the evolution of AngularJS, an open-source platform for front-end development. It helps create dynamic single page applications (SPAs). Also, it is based on features, like two-way binding, modularization, templating, mobile support, dependency injection, material design, and reactive programming.

Angular Setup,Install

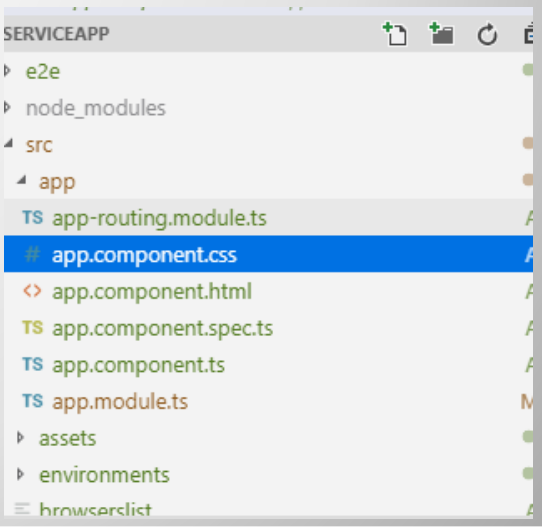
- Install Node.js from here
  - After downloading Node.js, the node package manager (npm) should automatically be installed. Test it out by doing:
  - $ npm --version
- Install Angluar CLI via npm:
  - $ npm install -g @angular/cli
- Navigate to project directory:
  - $ cd ~/reena/ $ mkdir appDir && cd appDir $ ng new cfe-app
- ng new takes a minute to run.
- Navigate to project & run local server
  - $ cd /path/to/your/newly/created/app/
  - $ cd ~/reena/appDir/cfe-app/ $ ng serve --open
- This will automatically open http://localhost:4200/
- *note: ng serve command launches the server, watches your files for changes, and rebuilds the app as you save changes*

Angular Component

The most basic building block of your Angular application . Components are a logical piece of code for Angular JS application. A Component consists of the following −

- **Template** − This is used to render the view for the application. This contains the HTML that needs to be rendered in the application. This part also includes the binding and directives.
- **Class** − This is like a class defined in any language such as C. This contains properties and methods. This has the code which is used to support the view. It is defined in TypeScript.
- **Metadata** − This has the extra data defined for the Angular class. It is defined with a decorator.

- . A component consists of three primary elements:
  - The HTML template
  - The logic
  - The styling (CSS, Sass, Stylus, etc..)



Angular Component

While we have three files here that represent the three elements above, the .ts (TypeScript) is the heart of the component. Let's take a look at that file:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'ng7-pre';
}
```

Here, because this is a component, we're importing Component from the @angular/core library, and then it defines what's called a @Component decorator, which provides configuration options for that particular component.

As you can see, it's referencing the location of the HTML template file and the CSS file with the templateUrl property and the styleUrls property.

The logic of the component resides in the class at the bottom. As you can see, the CLI starter template simply defines a single property called title.

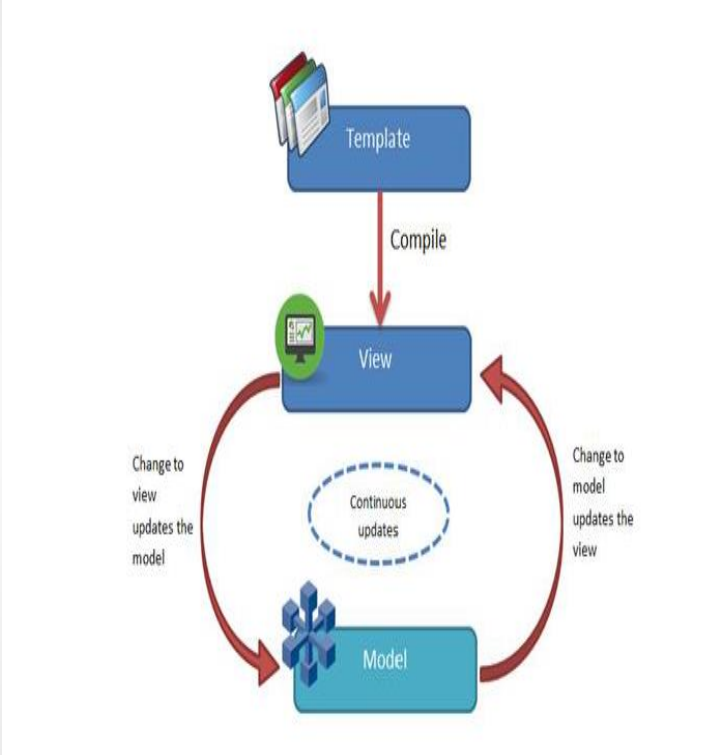
Angular CLI to create our own components

```
ng generate component nav
--or
ng g c nav
```

- Notice we first use the full syntax to generate a component, and then we use a shorthand syntax, which makes life a little bit easier. The commands do the same thing: generate components.
- When we generate components this way, it will create a new folder in the /src/ folder based on the name we gave it, along with the respective template, CSS, .ts and .spec (for testing) files.

DATA BINDING

- Data-binding in Angular apps is the automatic synchronization of data between the model and view components.
- The view is a projection of the model at all times. Data binding is the process that establishes a connection between the application UI and business logic.
- If the binding has the correct settings and the data provides the proper notifications, then, when the data changes its value, the elements that are bound to the data reflect changes automatically.



DIFFERENT TYPES OF DATA BINDING

- Interpolation
- Property Binding
- Two Way Binding
- Event Binding

Interpolation

- Interpolation data binding is the most popular and easiest way of data binding in angular.
- Interpolation use the braces expression i.e. {{ }} for render the bound value to the component template.
- It can be a static string or numeric value or an object of your data model. As per example, in angular we use it like below

```
{{firstName}}
```

PROPERTY BINDING

- Property binding used [] to send the data from the component to the html template.
- The most common way to use property binding is that assign any property of the html element tag into the [] with the component property value, i.e.

```
<input type="text" [value]="data.name"/>
```

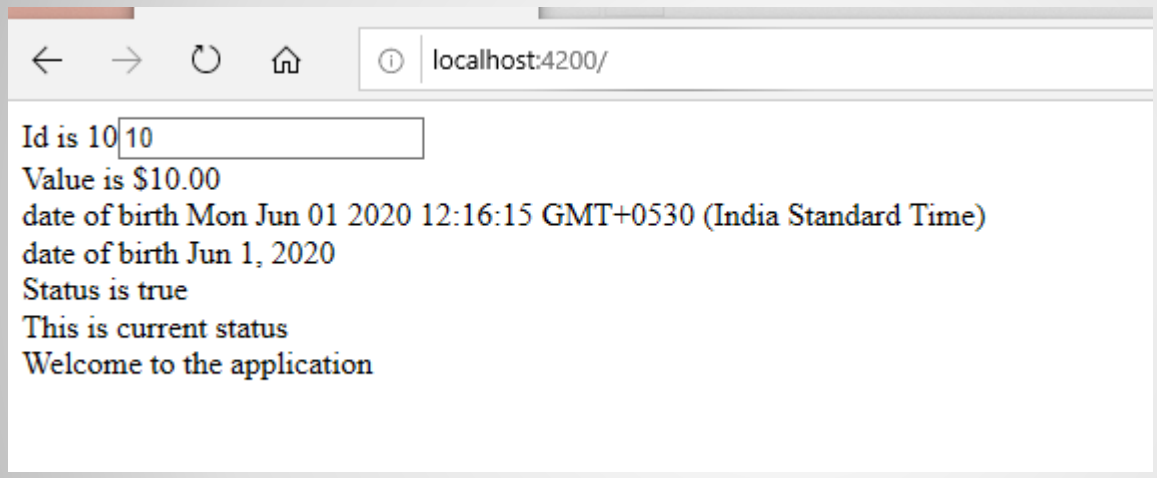
Defining Variable & functions in typescript file

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-bindingcomponent',
  templateUrl: './bindingcomponent.component.html',
  styleUrls: ['./bindingcomponent.component.css']
})
export class BindingcomponentComponent implements OnInit {
    value1:number=10;
    marks:Array<number>=[10,22,14];
    dob:Date=new Date();
    status:boolean=true;
    constructor() { }
    ngOnInit(): void {
   }
   returnString()
   {
     return "Welcome to the application";
   }
}
```

Interpolation & Property Binding in html

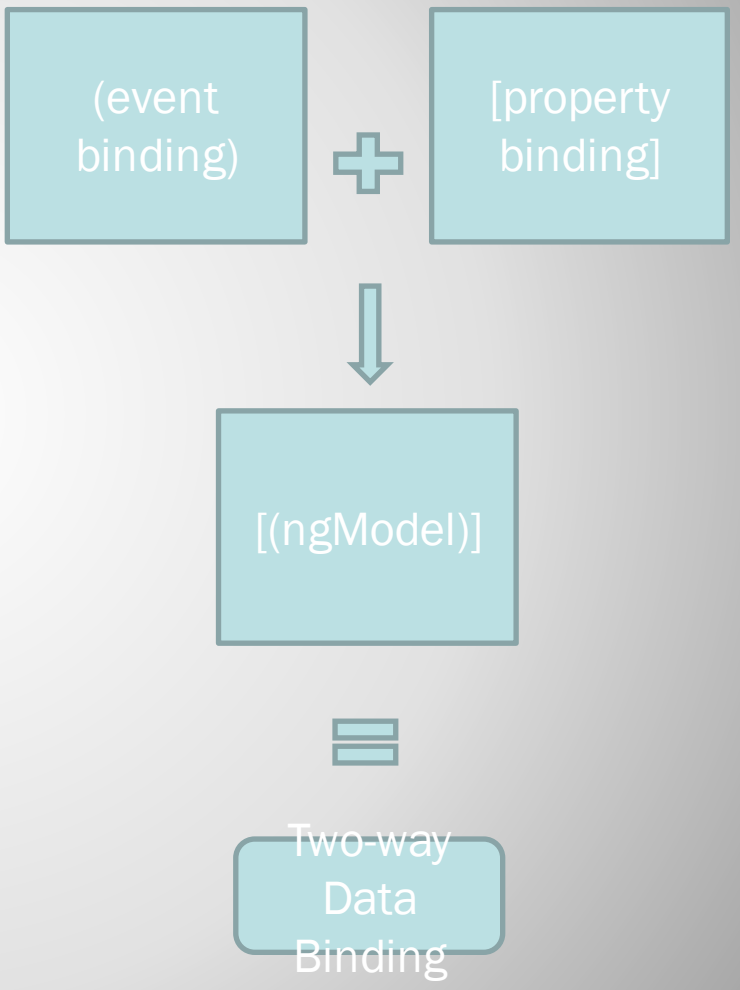
```
<div>
<span>Id is {{value1}}</span>
<input [value]="value1">
<br>
<span>Value is {{value1 | currency}}</span>
<br>
<span>date of birth {{dob}}</span>
<br>
<span>date of birth {{dob |date}}</span>
<br>
<span>Status is {{status}}</span>
<br>
<span> {{ status ? "This is current status":"This is false status
"}}</span>
<br>
<span>{{returnString()}}</span>
</div>
```

Output


localhost:4200/
Id is 10 10
Value is $10.00
date of birth Mon Jun 01 2020 12:16:15 GMT+0530 (India Standard Time)
date of birth Jun 1, 2020
Status is true
This is current status
Welcome to the application

Two way Property Binding

- The most popular and widely used data binding mechanism in two-way binding in the angular framework.
- Basically two-way binding mainly used in the input type field or any form element where user type or provide any value or change any control value in the one side, and on the other side, the same automatically updated in to the controller variables and vice versa.
- we use [] since it is actually a property binding and parentheses is used for the event binding concept.



Example

ts code

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-twowaybinding',
  templateUrl: './twowaybinding.component.html',
  styleUrls: ['./twowaybinding.component.css']
})
export class TwowaybindingComponent implements OnInit
{
            name:string="";
       constructor() { }

       ngOnInit(): void
       {
       }
}
```

html code

```
<input [(ngModel)]="name" type="text">

<span>Your name is {{name}}</span>
```

Following Error Occur after executing above

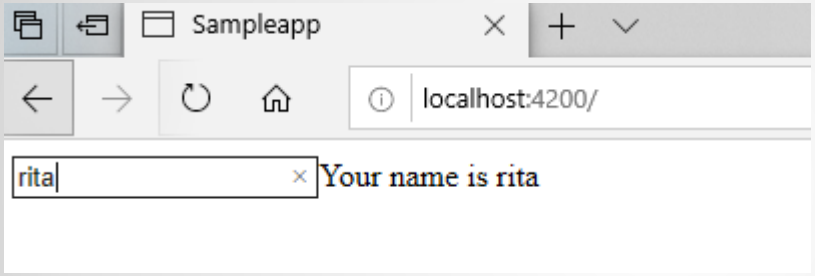
```
chunk {styles} styles.js, styles.js.map (styles) 9.7 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 339 kB [initial] [rendered]
Date: 2020-06-01T07:06:24.231Z - Hash: 294c0986356c2f8cef27 - Time: 7891ms

ERROR in src/app/twowaybinding/twowaybinding.component.html:1:8 - error NG8002: Can't bind to 'ngModel' since it isn't a kno
wn property of 'input'.

1 <input [(ngModel)]="name" type="text">
          ~~~~~~~~~~~~~~~~~~~~~~~~~~

  src/app/twowaybinding/twowaybinding.component.ts:5:16
    5   templateUrl: './twowaybinding.component.html',
                     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Error occurs in the template of component TwowaybindingComponent.
```

Resolve Above error

- Import FormsModule in app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import {FormsModule} from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BindingcomponentComponent } from './bindingcomponent/bindingcomponent.component'
;
import { TwowaybindingComponent } from './twowaybinding/twowaybinding.component';

@NgModule({
  declarations: [
    AppComponent,
    BindingcomponentComponent,
    TwowaybindingComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
```

Output is:



EVENT BINDING

- Event Binding is one of new mechanism which introduced by angular in a new concept.
- So, in angular , with properties we use square brackets and in events we use parenthesis.

```
<button (click)="showAlert();">Click</button>
```

Event Binding Example

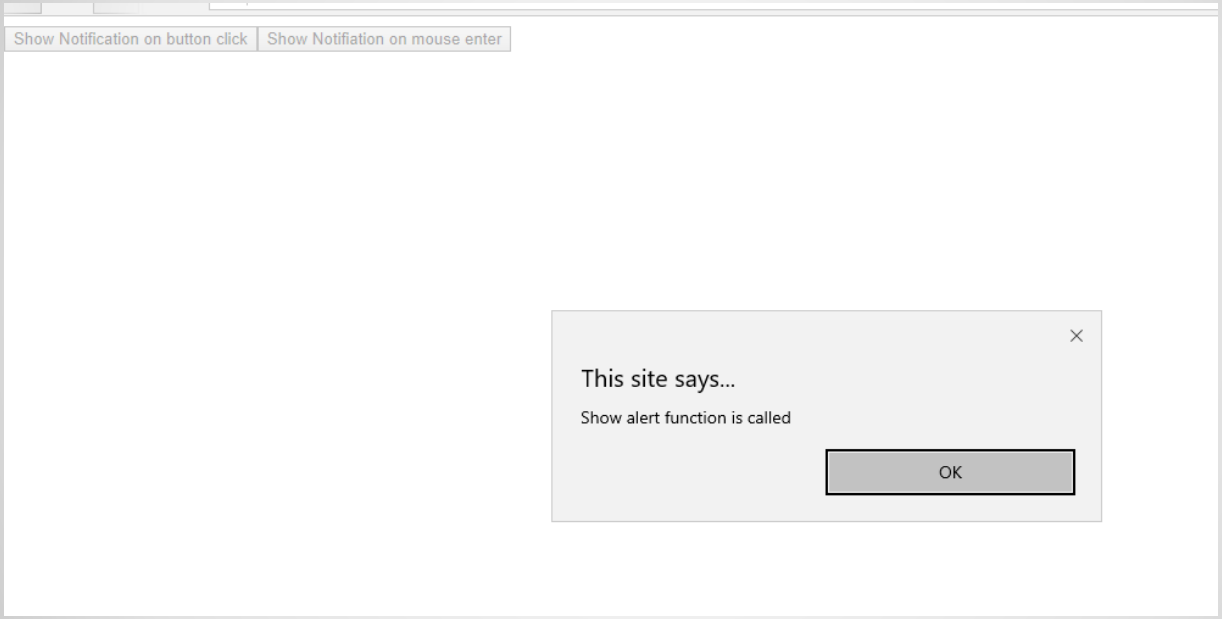
ts code

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-eventbinding',
  templateUrl: './eventbinding.component.html',
  styleUrls: ['./eventbinding.component.css']
})
export class EventbindingComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }
showAlert()
{
  alert("Show alert function is called");
}
}
```

Html code

```
<div>
    <button (click)="showAlert()">Show Notification on button click</button>
    <button (mouseenter)="showAlert()">Show Notifiation on mouse enter</button>
</div>
```

Output


Show Notification on button click
Show Notifiation on mouse enter
This site says...
Show alert function is called
OK

Recap

- Angular setup/installation
- Understanding Project structure
- What is Component
- Different Bindings in Angular
  - Interpolation
  - Property Binding
  - Two Way Binding
  - Event Binding

Web Resource

- https://angular.io/api/core/Component#:~:text=Components%20are%20the%20most%20basic,given%20element%20in%20a%20template.
- https://angular.io/guide/binding-syntax