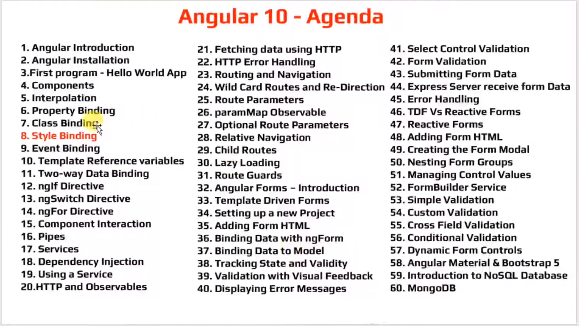
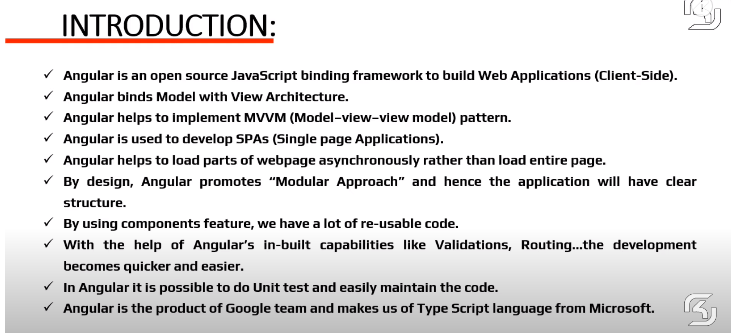
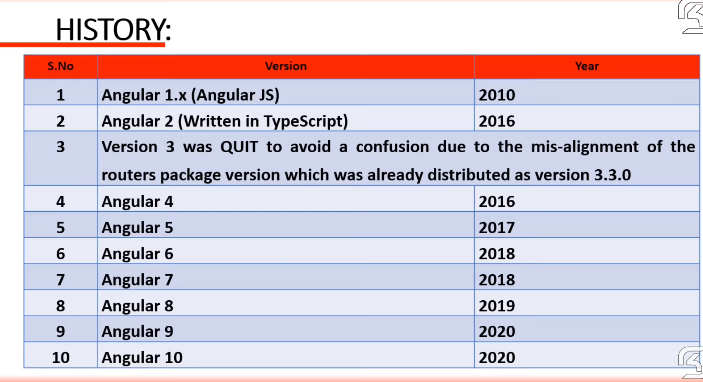
nRAK Helpline

<https://www.youtube.com/watch?v=Ebk8od4y1po&list=PLKIBeLKGEbp4cZdaLtOAVw8BH5VHXC3fT>

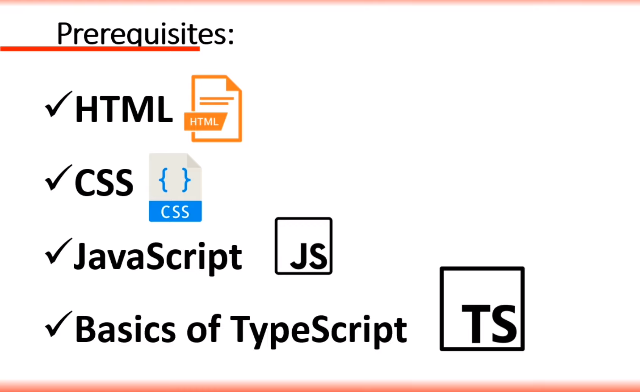
Angular10:

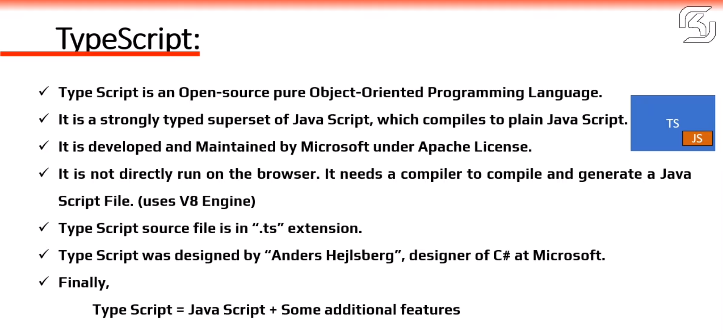


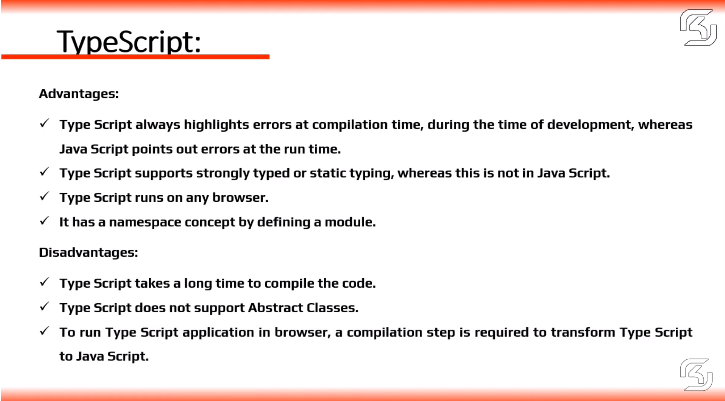










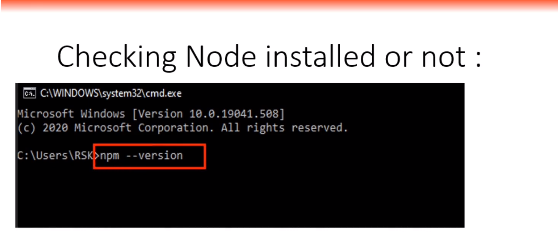


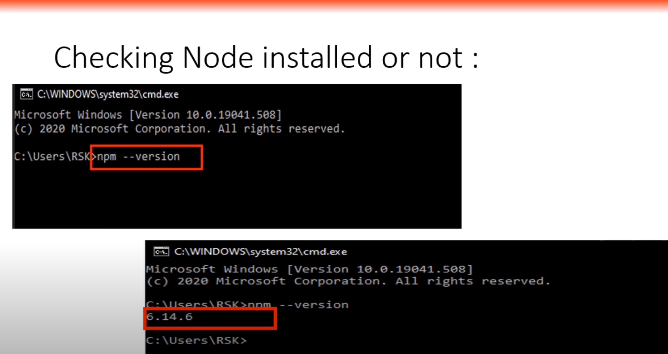
**2.Installation**

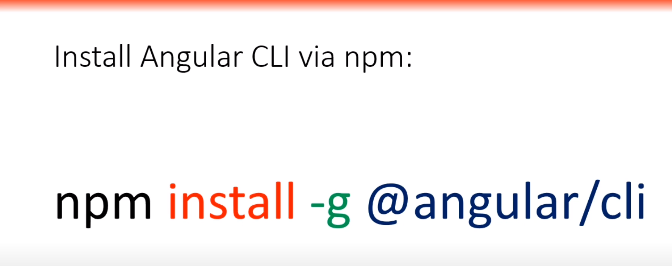
****

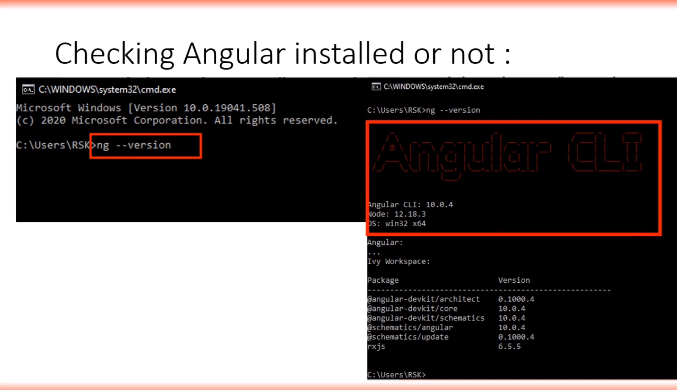
**npm (**Node Package Manager)

CLI (Command Line Interface)

****

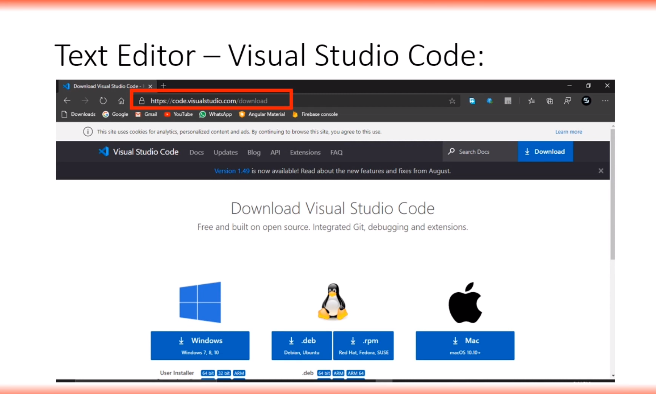
****

****

****

**Angular version check command is (version 16)**

**>ng version**

****

Creating new project?

How to open Integrated terminal

Open visual studio open new terminal type

>ng new Demo

It will ask Would you like to add Angular routing?

As of now it is not requried Press N

Which stylesheet format would you like to use?

Select CSS press Enter

It will take some time don’t warri your project will create. (this is dependent on your internet)

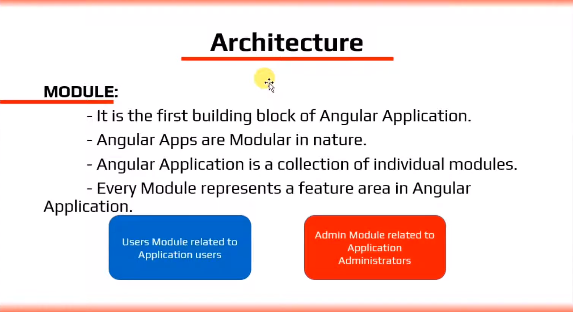
Then Run your project first you ne to move respect folder like Demo using

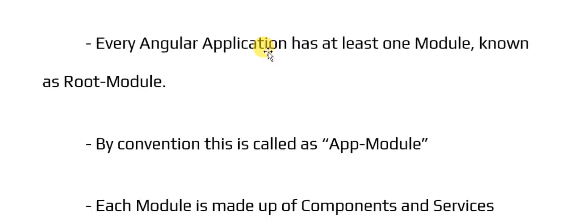
>cd Demo

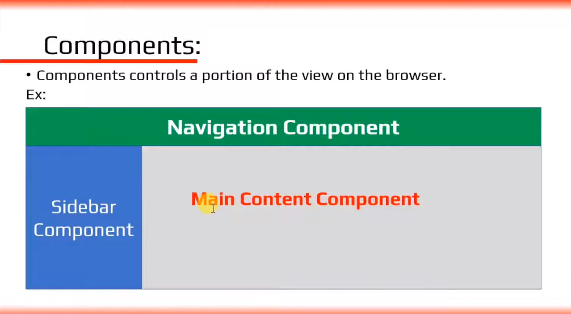
The Run the program

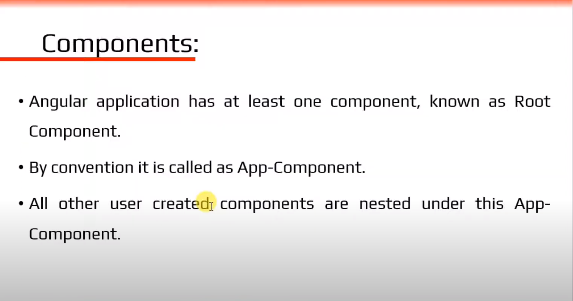
>ng serve

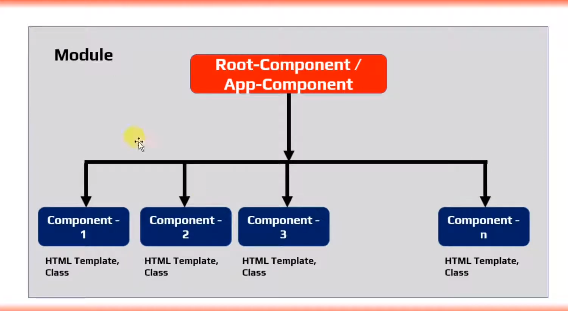
If you get any query press No

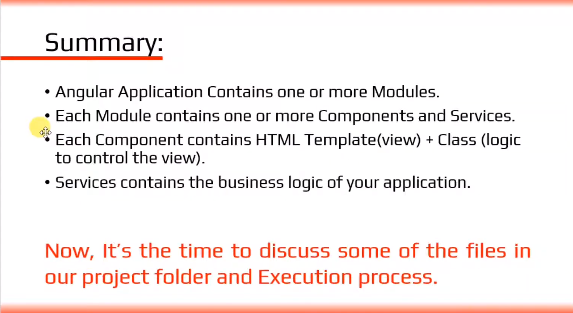
****

****

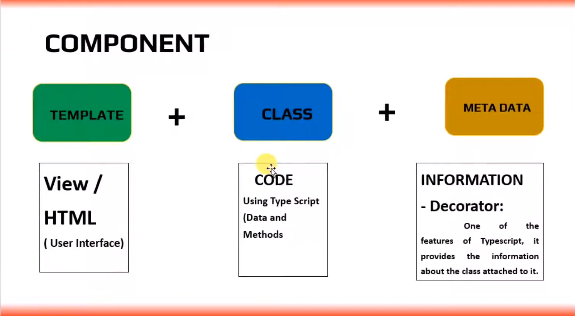
****

****

****

****

**4.Components**



User define component

>ng g c student

16 version

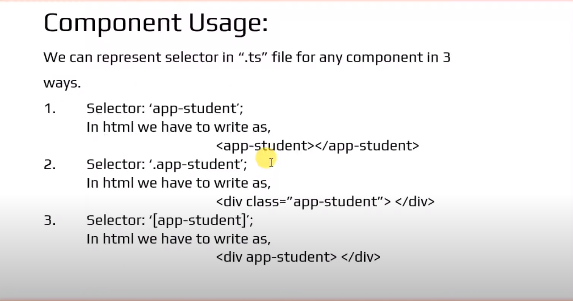
>ng g c student-details -t -s --skip-tests (press Enter)

-it(inline templet) –is(inline styles)

If we create component, it is automatically imported in **app.module.ts** and Import section and declaration section **StudentComponent** will be there

We will use 3 types of components

1. 1st one component is CUSTOM component that is recomended



Programs:

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

@Component({

  selector: 'app-test',

  template: `

    <p>

      Welcome to {{name}}

    </p>

    <p>

      {{"concatnation of "+name}}

    </p>

    <p>

      {{name.length}}

    </p>

    <p>{{name.toUpperCase()}}</p>

    <p>{{m1()}}</p>

  `,

  styles: [`

  p{

    text-align:center;

    font-size:40px;

    color:blue;

  }

  `

  ]

})

export class TestComponent {

  public name="Angular 10 Interpolation";

  m1(){

    return "welcome" +this.name;

  }

}

**5.Interpolation**

Useage of Interpolation:

Dynamic content display perpose we will using interpolation.

Oneway binding:

It move to class to 🡪 view.

It will not go view to 🡪 class.

What not to do:

Assinement dot declare

Global variables dot declare

* If you want declare global variables create variable in class level and declare on view

Create a new component including html,css and skip the Spect file command.

>ng g c test –it –is –skipTests=true

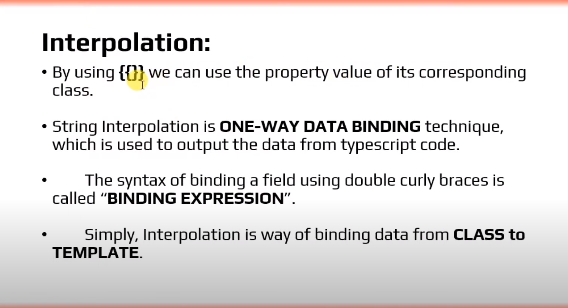
-it(inline templet) –is(inline styles)

If it is not work Use Help commend the write like

>ng g c test –t –s --skip-tests=true (I m not sure this command will work or not)

Use this Command in 16 version (this command will work)

>ng g c test –t –s –skip-tests (press Enter)

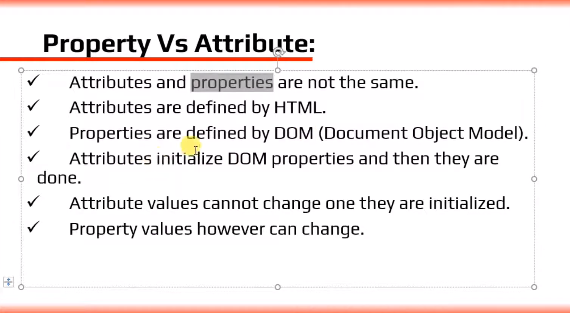


**6.Property Binding:**

$0(dollar zero means current element)

Q)can we property Binding using interpolation.?

A) yes we can



Program:

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

@Component({

  selector: 'app-test',

  template: `

    <br><br>

      <input [id]="courseId" type="text" value="Angular10"><br><br>

      <input [disabled]="isDisabled" id="{{courseId}}" type="text" value="Angular10"><br><br>

  `,

  styles: [

  ]

})

export class TestComponent {

  public courseId="123";

  public isDisabled=true;

}

**7.Class Binding:**

Program:

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

@Component({

  selector: 'app-test',

  template: `

    <h2>Class Binding</h2>

    <h2 class="textcolor">class binding</h2>

    <h2 [class]="mycolor1">color Binding</h2>

    <h2 [class]="mycolor2">Size Binding</h2>

    <h2 [class]="mycolor3">Style Binding</h2>

    <br><br>

    <h2 [class.textSize]="requried">Conditinal Binding</h2>

    <h2 [ngClass]="Group">Conditinal Binding</h2>

  `,

  styles: [`

  h2{

    text-align:center;

    font-size:30px;

    color:gold;

  }

  .textColor{

    color:green;

  }

  .textSize{

    font-size:60px;

  }

  .textStyle{

    font-style:italic;

  }

  `

  ]

})

export class TestComponent {

  public mycolor1="textColor";

  public mycolor2="textSize";

  public mycolor3="textStyle";

  public requried=true;

  public Group ={

    textColor:this.requried,

    textSize:this.requried,

    textStyle:this.requried,

  }

}

**8.Style Binding:**

Program:

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

@Component({

  selector: 'app-test',

  template: `

    <h2>Style Binding</h2>

    <h2 [style.color]="'blue'">Angular Style Binding</h2>

    <h2 [style.color]="mycolor">Style Binding</h2>

    <h2 [style.color]="req ? 'green' : 'red'">Conditinal Style Binding</h2>

    <h2 [ngStyle]="groupStyle">Object Styeling</h2>

  `,

  styles: [`

    h2{

      text-align:center;

    }

  `

  ]

})

export class TestComponent {

    public mycolor="orange";

    public req=false;

    public groupStyle={

      fontStyle:"italic",

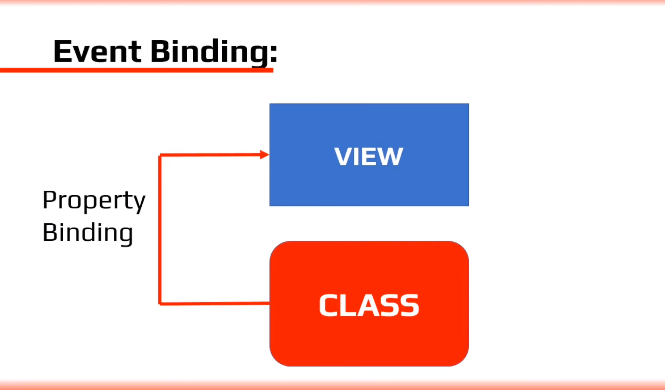
      color:"red",

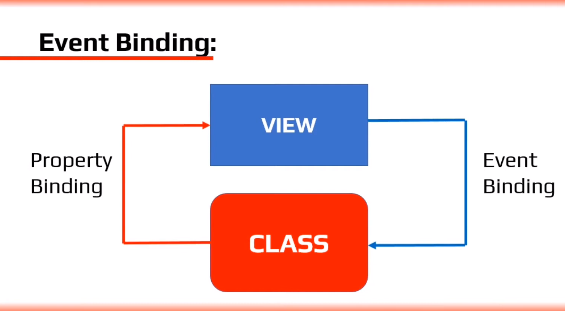
      fontSize:"40px"

    }

}

**9.Event Binding:**







Program:

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

@Component({

  selector: 'app-test',

  template: `

    <h2>Event Binding</h2>

    <button (click)="onclick()" type="button">click me</button>

    <h2>{{greeting}}</h2>

  `,

  styles: [`

    h2{

      text-align:center;

    }

  `

  ]

})

export class TestComponent {

      public greeting="";

      onclick(){

        // console.log("Welcome to Angular");

        this.greeting="Welcome to Event Binding using interpolation";

      }

  }

**10.Template Reference Variables:** ( fasing sum problem in class -> message)

Program:

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

@Component({

  selector: 'app-test',

  template: `

          <h2>Template Reference Variables</h2>

          <br>

          // <input type=text>

          // <button type="button">Welcome</button>

          <input #msg type="text">

          <button (click)="message(msg.value)" type="button">Welcome</button>

  `,

  styles: [`

    h2{

      text-align:center;

    }

  `

  ]

})

export class TestComponent {

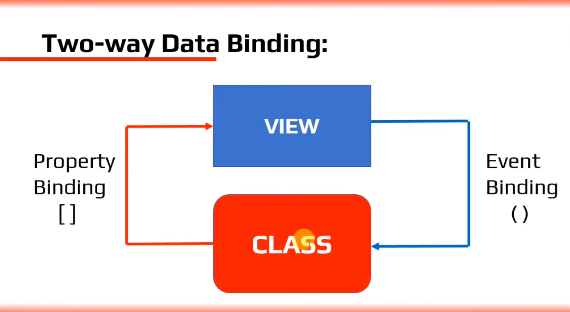
  message(message){

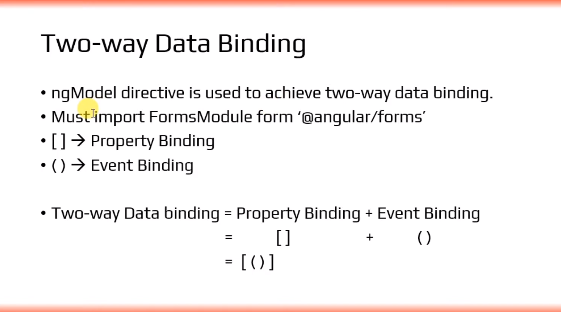
    console.log(message);

  }

  }

**11.Two way Data Binding:**





Program:

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

@Component({

  selector: 'app-test',

  template: `

          <h2>Two way Data Binding</h2>

          <br>

          <br>

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

          <button typ="button">click me</button>

          <br>

          <br>

          {{data}}

  `,

  styles: [`

    h2{

      text-align:center;

    }

  `

  ]

})

export class TestComponent {

    public data="";

  }

--------------

In app.Module You need to import sum values

app.Module.ts

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

 imports: [

    BrowserModule,

    FormsModule

  ],

# 12. \*NgIf in Angular:

# 

# Program:

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

@Component({

  selector: 'app-test',

  template: `

          <h2>\*ngIf in Angular</h2>

          <br>

          // <h2 \*ngIf="true">{{name}} in Angular application</h2>

          <h2 \*ngIf="status; then trueBlock else elseBlock"></h2>

          <ng-template #elseBlock>

             <h2> {{name}}==> else block</h2>

          </ng-template>

          <ng-template #trueBlock>

              <h2>{{name}} ==> trueBlock</h2>

          </ng-template>

  `,

  styles: [`

    h2{

      text-align:center;

      color:blue;

    }

  `

  ]

})

export class TestComponent {

    public status=true;

    public name="Welcome to ngIf";

  }

# 13.ngSwitch:

# Program

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

@Component({

  selector: 'app-test',

  template: `

          <h2>ngSwitch in Angular</h2>

          <div [ngSwitch]= "myChoice">

            <div class="switchCase">

              <div \*ngSwitchCase="'one'">First Switch Block</div>

              <div \*ngSwitchCase="'two'">Second Switch Block</div>

              <div \*ngSwitchCase="'three'">Third Switch Block</div>

              <div \*ngSwitchDefault>Invalid Choice</div>

            </div>

          </div>

            `,

  styles: [`

    h2{

      text-align:center;

      color:blue;

    }

    .switchCase{

      text-align:center;

      font-size:40px;

      color:gold;

      font-width:bold;

    }

  `

  ]

})

export class TestComponent {

    public myChoice="two";

  }

# 14.ngFor Directive:

# Program:

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

@Component({

  selector: 'app-test',

  template: `

          <h2>ngFor Directive</h2>

          <div \*ngFor="let x of myChoice; index as i; last as l;first as f;even as e;odd as o">

            <h2>{{i}} ===> {{x}} ===>{{l}} ==>{{f}} ==>{{e}} ==>{{o}}</h2>

          </div>

            `,

  styles: [`

    h2{

      text-align:center;

      color:blue;

    }

  `

  ]

})

export class TestComponent {

    public myChoice=["Angular","React","Bootstrap","Typescript"];

  }

# 15.Component Interaction: (fasing problem parent to chaild we canot import)

# 

# 16.Pipes:

# Q) what is pipes.?

# Q) what are the different types of pipes.?

# Q) what is mean by pipes.?

# A) Pipes are use to transform the incoming stream of data into required format before displaing into view

# Program:

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

@Component({

  selector: 'app-test',

  template: `

          <h2>{{name}}</h2>

          <h2>{{name | lowercase}}</h2>

          <h2>{{name | uppercase}}</h2>

          <h2>{{msg | titlecase}}</h2>

          <h2>{{name | slice:3}}</h2>

          <h2>{{name | slice: 3:5}}</h2>

          <h2>{{person | json}}</h2>

          <br>

          <h2>{{5.678 | number: '1.2-3'}}</h2>

          <h2>{{5.678 | number: '3.4-5'}}</h2>

          <h2>{{5.678 | number: '3.1-2'}}</h2>

          <br>

          <h2> {{0.25 | percent}}</h2>

          <h2> {{0.25  | currency}}</h2>

          <h2> {{0.25  | currency: 'INR'}}</h2>

          <br>

          <h2>{{date}}</h2>

          <h2>{{date | date: 'short'}}</h2>

          <h2>{{date | date: 'shortDate'}}</h2>

          <h2>{{date | date: 'shortTime'}}</h2>

          <h2>{{date | date: 'medium'}}</h2>

          <h2>{{date | date: 'mediumDate'}}</h2>

          <h2>{{date | date: 'mediumTime'}}</h2>

          <h2>{{date | date: 'long'}}</h2>

          <h2>{{date | date: 'longDate'}}</h2>

          <h2>{{date | date: 'longTime'}}</h2>

            `,

  styles: [`

    h2{

      text-align:center;

      color:blue;

      background-color:black;

    }

  `

  ]

})

export class TestComponent {

  public name="PIPESIn Angular";

  public msg="Welcome to Pipe Consept";

  public person ={

    "firstName":"Angular",

    "secondName":"React"

  }

       public date =new Date();

  }

# 17.Services: (Varry Important)

# 

# 

# Program:

# App.component.html

<app-student-list></app-student-list>

<app-student-details></app-student-details>

# Student-list

# 

# Student-list.component.ts

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

@Component({

  selector: 'app-student-list',

  template: `

    <h1>

      student-list works!

    </h1>

    <ul \*ngFor="let x of stuli">

      <li> {{x.name}}</li>

    </ul>

  `,

  styles: [

  ]

})

export class StudentListComponent {

    public stuli=[

      {"id":1,"name":"Rama","course":"Angular"},

      {"id":2,"name":"laxman","course":"React"},

      {"id":3,"name":"Sita","course":"Bootstrap"},

      {"id":4,"name":"Ravana","course":"Typescript"}

    ];

}

# Student-details

# Student-details.component.ts

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

@Component({

  selector: 'app-student-details',

  template: `

    <h1>

      student-details works!

    </h1>

    <ul \*ngFor="let x of stude">

      <li>{{x.id}} --> {{x.name}} -->{{x.course}}</li>

    </ul>

  `,

  styles: [

  ]

})

export class StudentDetailsComponent {

  public stude=[

    {"id":1,"name":"Rama","course":"Angular"},

    {"id":2,"name":"laxman","course":"React"},

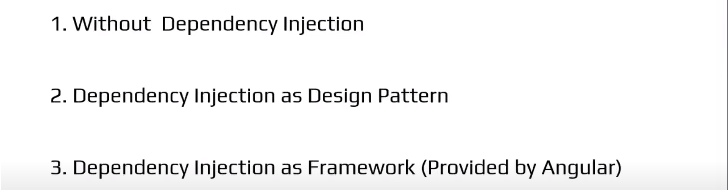
    {"id":3,"name":"Sita","course":"Bootstrap"},

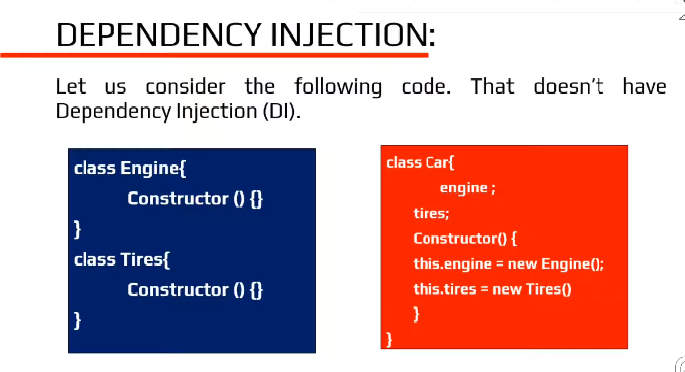
    {"id":4,"name":"Ravana","course":"Typescript"}

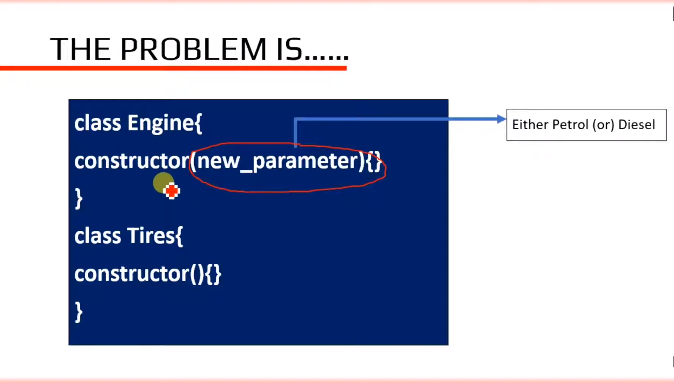
  ];

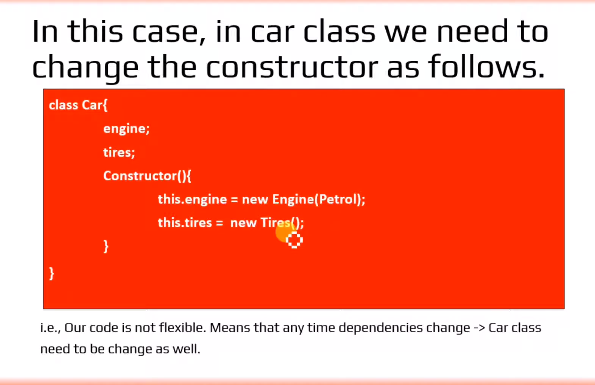
}

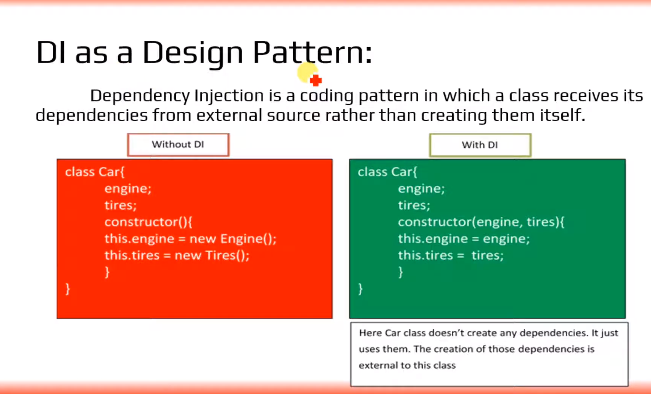
**18.Dependency Injection:**

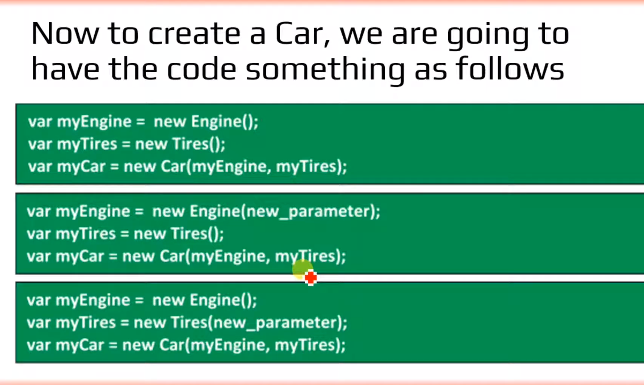






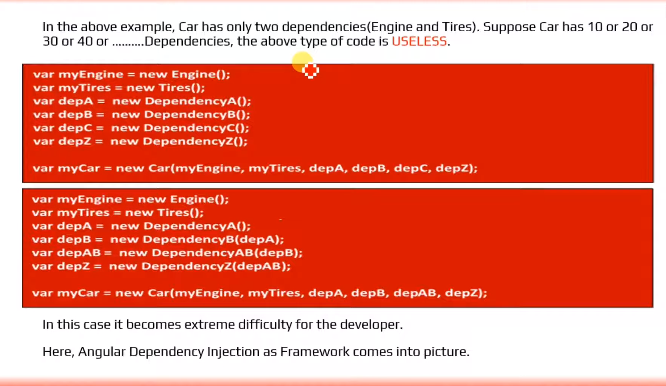


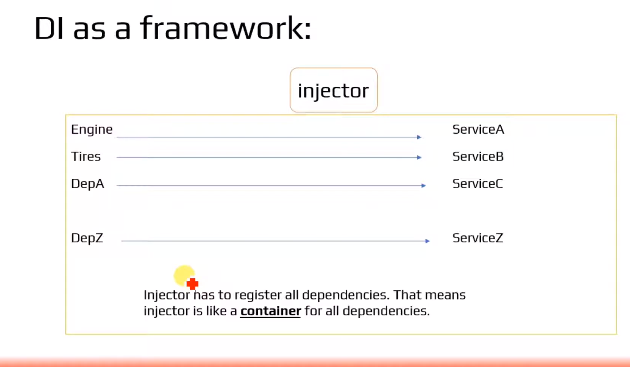




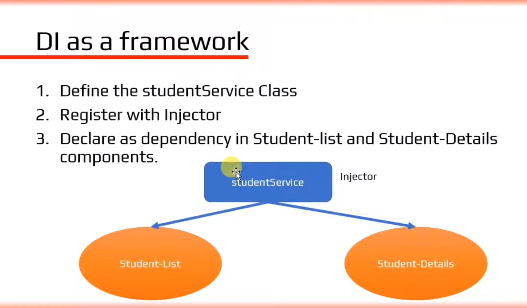
Problem solved?

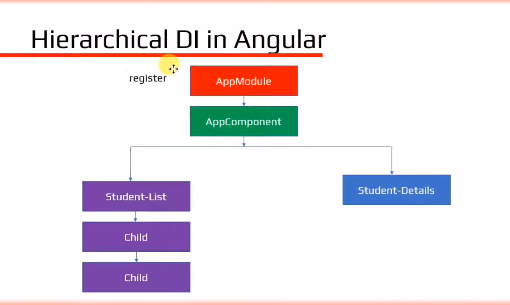
No Another problem exits!



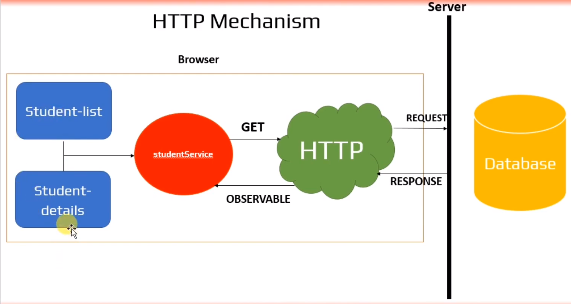


**19.Using a Service:** (Service Dipendancy Program was not work Import section was not created)

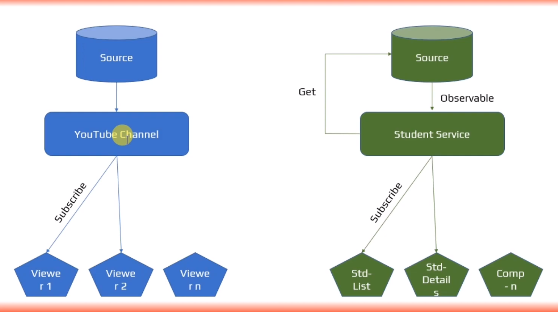


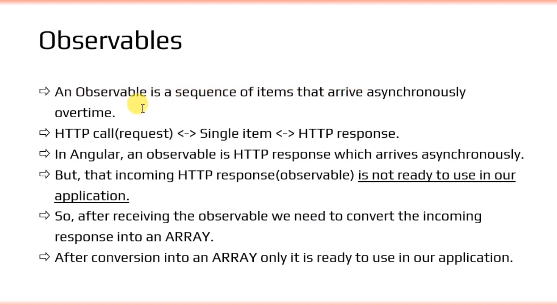


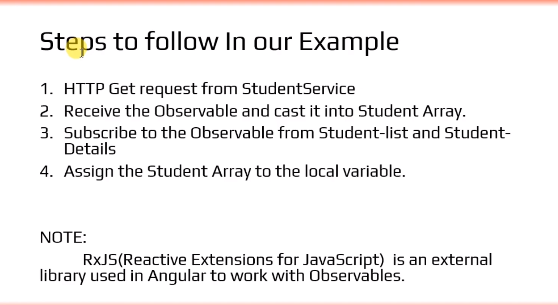
**20.HTTP & Observables: (20,21,22 class are dependent by one to one that’s why I’m unable to work)**



For Example



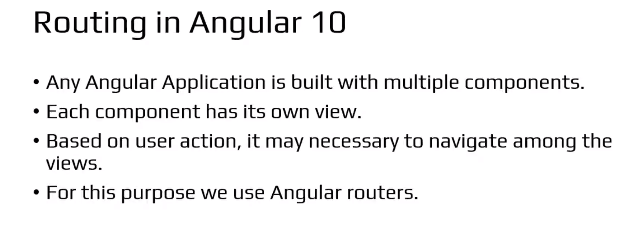


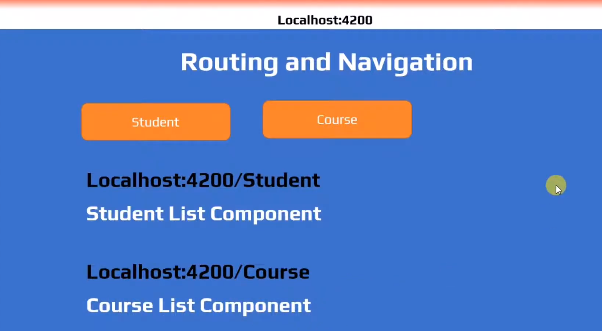


**21.Fetching data using HTTP:**

**22.HTTP Error Handling:**

**23.Routing and Navigation:**





Program:

Crate application while creating application it will ask routing permission press Yes(y)

And generate a two components

1.Student-List

2.Course-List

>ng g c student-list –t –s - -skip-tests=true (presenter)

> ng g c course-list –t –s - -skip-tests=true (presenter)

student-list.component.ts

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

@Component({

  selector: 'app-student-list',

  template: `

    <h2>

      Student-List Component

    </h2>

  `,

  styles: [`

  h2{

    text-align: center;

    color: blue;

    font-family: play;

    font-weight: bold;

    font-size: 30px;

}

  ` ]

})

export class StudentListComponent {

}

App-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

import { StudentListComponent } from './student-list/student-list.component';

import { CourseListComponent } from './course-list/course-list.component';

const routes: Routes = [

  {path:'student',component:StudentListComponent},

  {path:'Course',component:CourseListComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

export const myRouting=[

                          StudentListComponent,

                          CourseListComponent

];

Course-list.component.ts

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

@Component({

  selector: 'app-course-list',

  template: `

    <h2>

      Course-List Component!

    </h2>

  `,

  styles: [`

    h2{

      text-align: center;

      color:blue;

      font-family: play;

      font-weight: bold;

      font-size: 30px;

  }

  `

  ]

})

export class CourseListComponent {

}

App.module.ts

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

@NgModule({

  declarations: [

    AppComponent,

    myRouting

app.component.html

<h1>Angular 10 Application</h1>

<nav>

  <a routerLink="/student" routerLinkActive="active">Student-list</a>

  <a routerLink="/Course" routerLinkActive="active">Course-list</a>

</nav>

<router-outlet></router-outlet>

Style.css

/\* You can add global styles to this file, and also import other style files \*/

h1{

    padding: 0;

    margin: 0;

    font-family: play;

    font-size: 50px;

    font-weight: bold;

    text-align: center;

    color: red;

}

/\* Navigation link styles \*/

nav a {

    padding: 15px 20px;

    text-decoration: none;

    margin-top: 10px;

    display: inline-block;

    background-color: #eee;

    border-radius: 4px;

    margin-left: 20px;

  }

  nav a:visited, a:link {

    color: #3643b6;

  }

  nav a:hover {

    color: #fff;

    background-color: #3643b6;

  }

  nav a.active {

    color:  #ce2b2b;

    background-color: #49ce0c;

  }

**24.Wild Card Routes:**

Generating new component pagenot fount

>ng g c pate-not-fount –t –s - -skip-tests=true (press enter)

Wild card entery must be Last, if you use first it will show error

Program:

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

import { RouterModule, Routes } from '@angular/router';

import { StudentListComponent } from './student-list/student-list.component';

import { CourseListComponent } from './course-list/course-list.component';

import { PageNotFoundComponent } from './page-not-found/page-not-found.component';

const routes: Routes = [

  {path:'',redirectTo:'student',pathMatch:'full'},

  {path:'student',component:StudentListComponent},

  {path:'Course',component:CourseListComponent},

  {path:"\*\*",component:PageNotFoundComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

export const myRouting=[

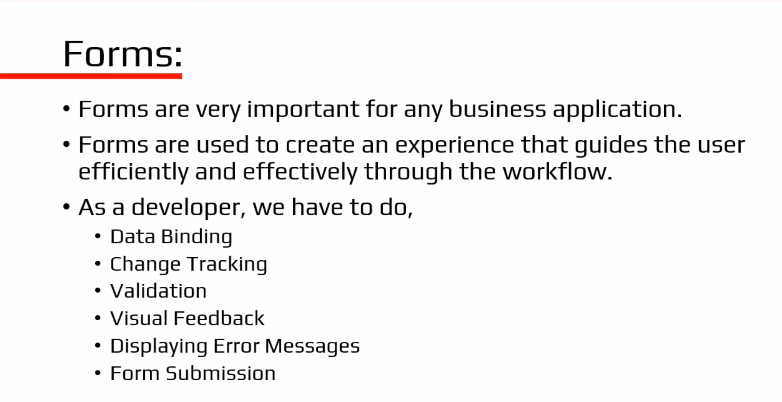
                          StudentListComponent,

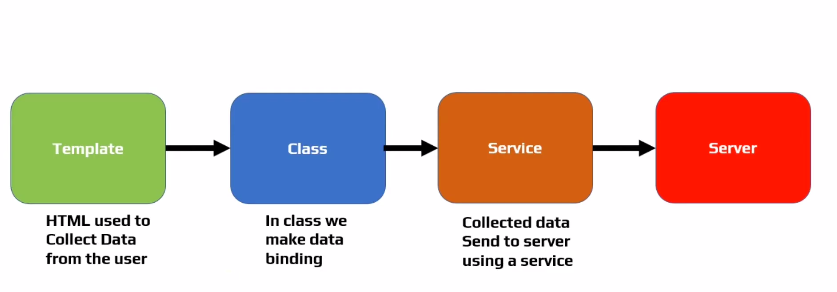
                          CourseListComponent,

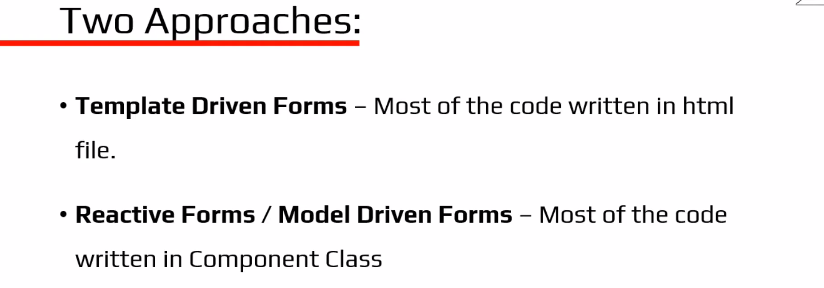
                          PageNotFoundComponent

];

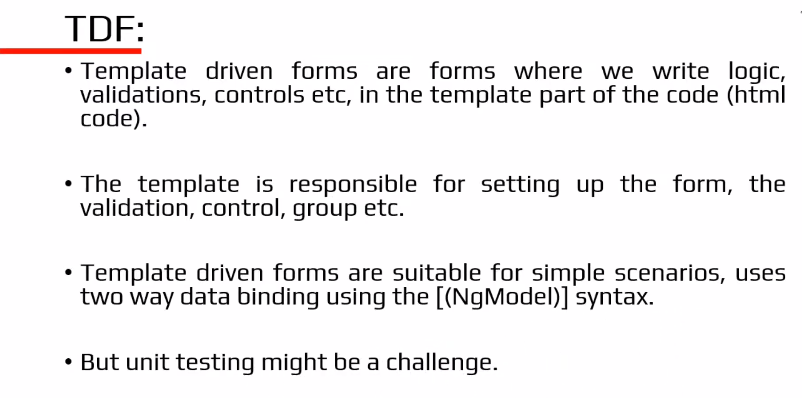
**32.Angular Forms – Introduction**

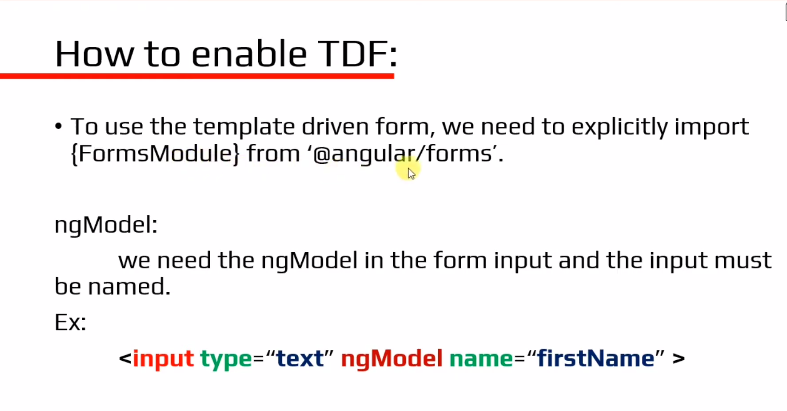
****

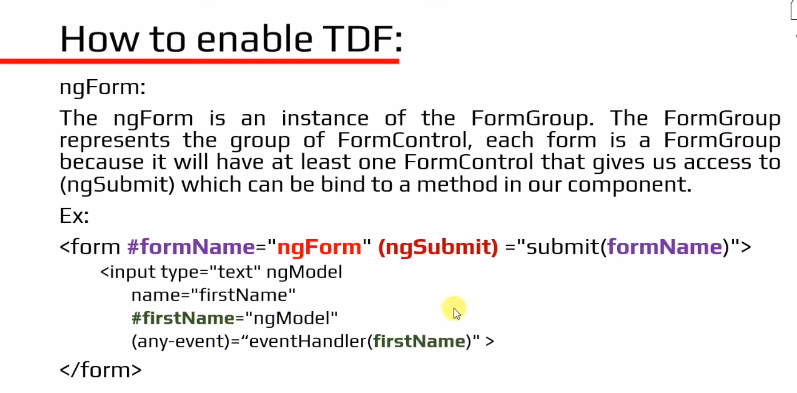




**33 Template Driven Forms Introduction**







**34 Setting Up Project with Bootstrap** (Bootstrap, jquery and popper is not installed ) (individual bootstrap and jquery installed)

How to create a new Application

>create a new folder(desktop) then open visual studio and click on “files” -> select “Open folder” then choose your location and select it your application is created

In Eggsting project how to install bootstrap

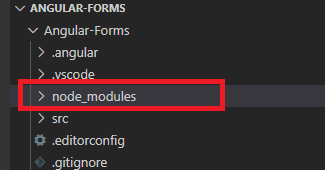
>npm install bootstrap jquery popper - - save (press Enter)(Here Bootstrap, jquery and popper installed)

Bootstrp installation command

> npm i bootstrap --save

How to check Bootstrap is installed or not ?

In a application node\_modules there open scroll it you will find “bootstrap” .



Ofter installing bootstrap you need to give path of bootstrap

Goto the “angular.json” there “**style**” ,”**script**” there, you need to give bootstrap and jquery path (location).

**“style”**

give the bootstrap path using double cotations Ex:-(“node\_modules/bootstrap/dist/css/bootstrap.min.css”)

**“script”**

**NOTE :**  first give jqery path(location) and bootstrap path

**Ex:-**

“node\_modules/jquery/dist/jquery.min.js”,

“node\_modules/bootstrap/dist/js/bootstrap.min.js”

Create one component

>ng g c tdf - - skip-tests=true

There bootstrap code

**Tdf.component.html**

<p>Welcome to Template driven forms</p>

<br>

<div class="form-group">

    <label>User Name:</label>

    <input type="text" class="form-control" name="username" placeholder="Enter your username">

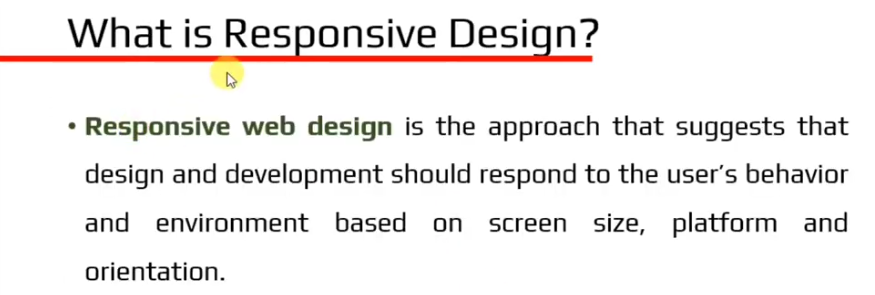
</div>

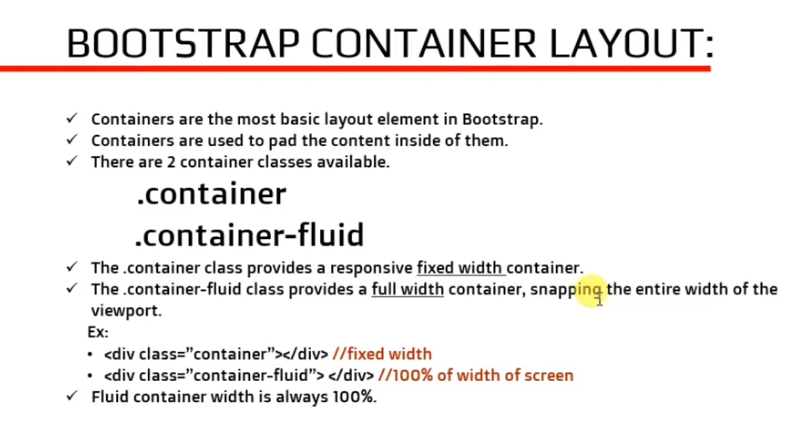
<br>

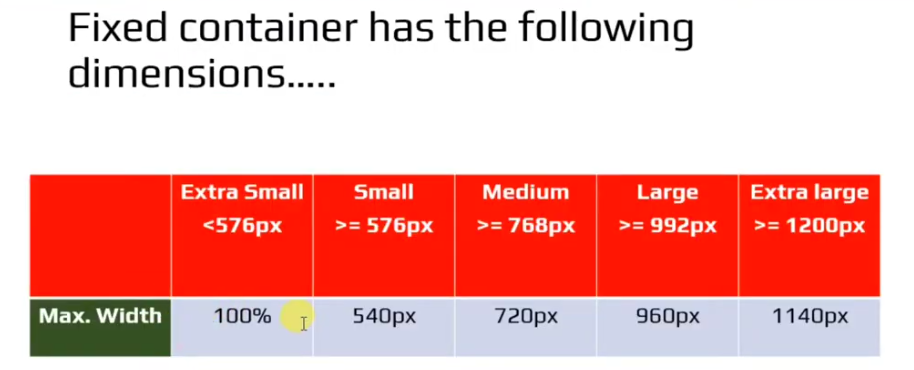
<button type="submit" class="btn btn-primary">submit</button>

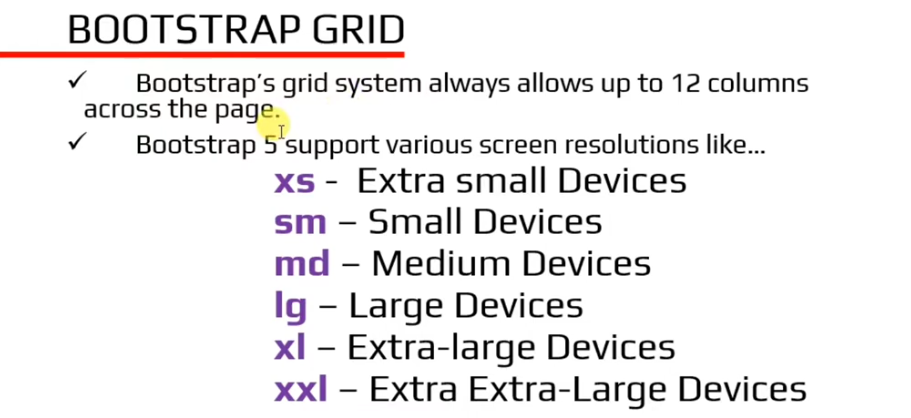
**35. Adding HTML Elements using Bootstrap**

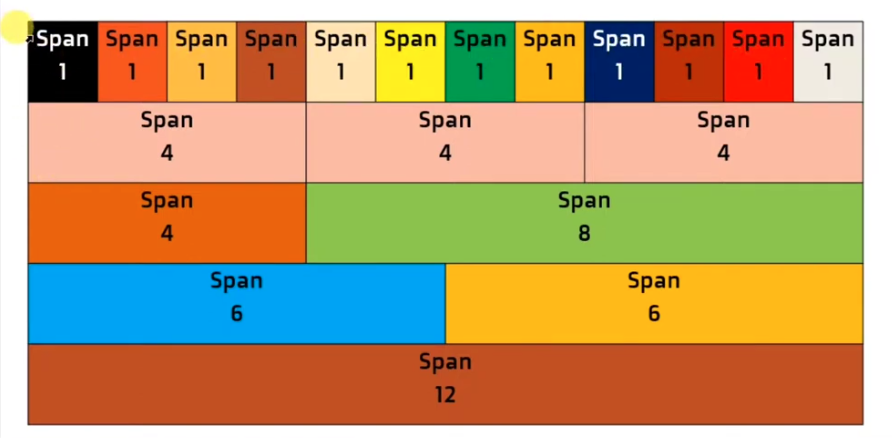












Program:

Tdf.component.ts

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

@Component({

  selector: 'app-tdf',

  templateUrl: './tdf.component.html',

  styleUrls: ['./tdf.component.css']

})

export class TdfComponent {

public course=["Angular","Bootstrap","React","MangoDM"];

}

Tdf.component.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <div class="container-fluid">

        <div class="row">

            <div class="col-6">

                <div class="form-group">

                    <label>Username</label>

                    <input type="text" class="form-control" placeholder="Username" aria-label="Username"

                        aria-describedby="basic-addon1">

                </div>

                <br>

                <div class="form-group">

                    <input type="email" class="form-control" placeholder="email" aria-label="email"

                        aria-describedby="basic-addon1">

                </div>

                <br>

                <div class="form-group">

                    <input type="tel" class="form-control" placeholder="Mobile number" aria-label="phone"

                        aria-describedby="basic-addon1">

                </div>

            </div>

            <div class="col-6">

                <div class="form-check form-check-inline">

                    <input class="form-check-input" type="radio" name="inlineRadioOptions" id="inlineRadio1"

                        value="option1">

                    <label class="form-check-label" for="inlineRadio1">Male</label>

                </div>

                <div class="form-check form-check-inline">

                    <input class="form-check-input" type="radio" name="inlineRadioOptions" id="inlineRadio1"

                        value="option1">

                    <label class="form-check-label" for="inlineRadio1">Female</label>

                </div>

                <div class="input-group">

                    <select class="form-select" id="inputGroupSelect01">

                        <option selected>Choose...</option>

                        <option \*ngFor="let x of course">{{x}}</option>

                    </select>

                </div>

                <br>

                <div class="form-check">

                    <input type="checkbox" class="form-check-input" value="" id="flexCheckDefault">

                    <label class="form-check-label" for="flexCheckDefault">

                        Default checkbox

                    </label>

                </div>

                <br>

                <div class="custom-control custom-checkbox">

                    <input type="checkbox" class="custom-control-imput" id="customCheckDisabaled1">

                    <label class="custom-control-label" for="customCheckDisabaled1"> Remember me</label>

                </div>

            </div>

        </div>

        <br>

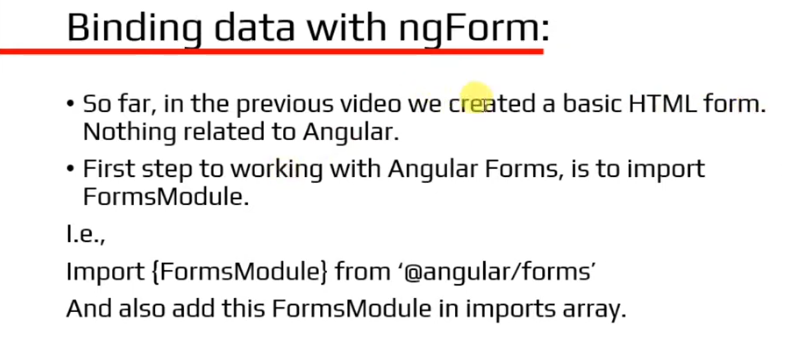
        <button type="submit" class="btn btn-primary">Register</button>

    </div>

</body>

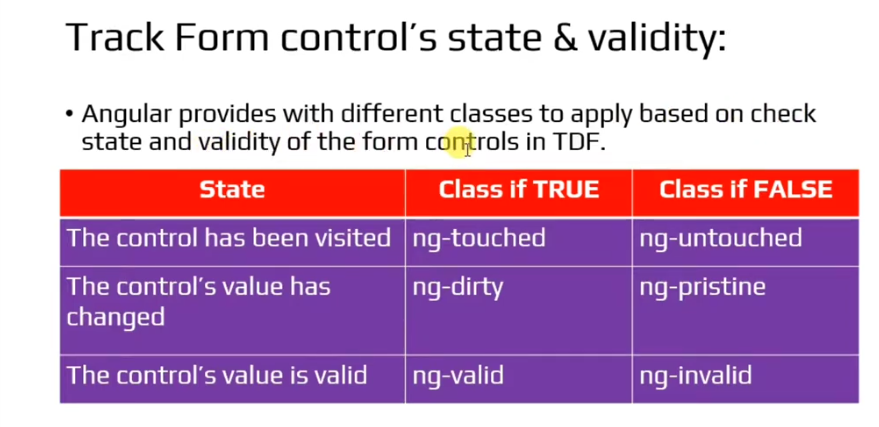
</html>

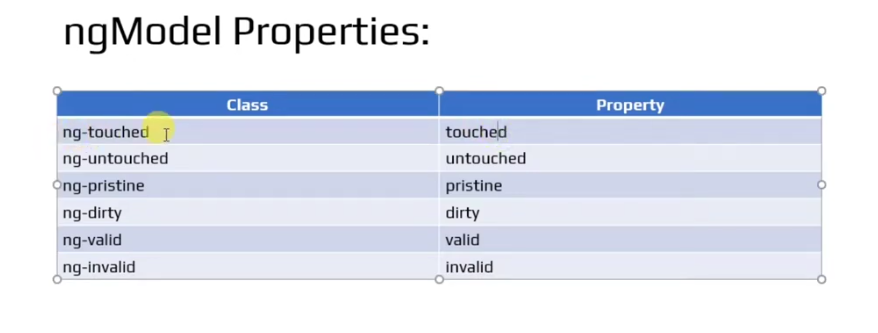
**36. Binding Data using ngForm**



**37. Bind Data to User Model**

**38 Tracking State & Validity**





**39 . Displaying Visual Feedback**

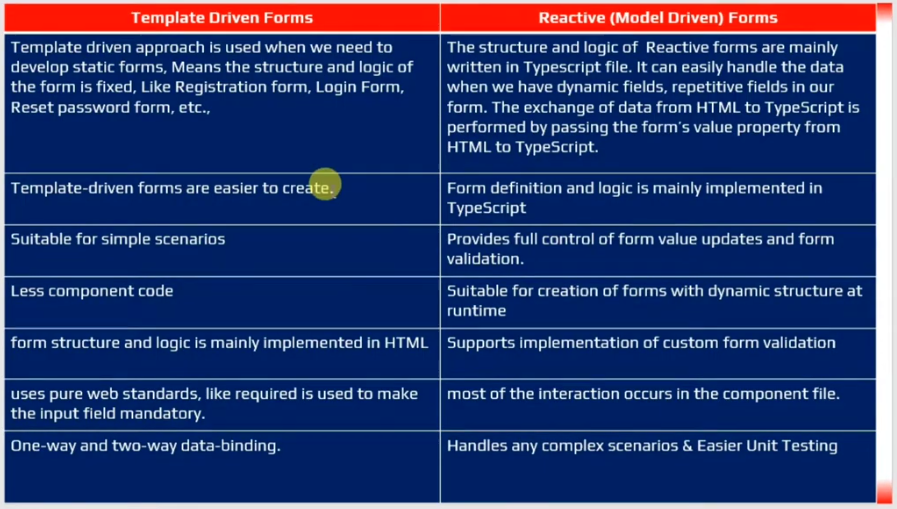
**40 . Displaying ERROR Messages**

**41 . Select Control Validation**

**42 . FORM Validation**

**43 . Submitting FORM Data**

**46 . Template Driven Forms Vs Reactive Forms**

****

**47 . Reactive Forms**

****

**48 . Reactive Forms- Adding HTML Form**

**Adding Forms HTML:**

Angular Unstaling

1.open command prompt and run command as given

2.>npm uninstall –g @angular/cli

3.>npm cache clean - -fource

Successfully uninstalled then check angular is there or not

>ng -version

Commands:

How to create a new Application

>create a new folder(desktop) then open visual studio and click on “files” -> select “Open folder” then choose your location and select it your application is created

Creating a new component

>ng g c student

Create a new component including html,css and skip the Spect file command.

>ng g c test –it –is –skipTests=true

16 version

>ng g c student-details -t -s --skip-tests (press Enter)

-it(inline templet) –is(inline styles)

Save all files in VS

>ctrl+k+s

Creating a Service

>ng g s (name of the servive)student --skit-tests (or) >ng g s (name of the servive)student --skit-tests=true

How to stop Server

>ctrl+c and press Y

**Bootstrap Install and uninstall commands**

npm uninstall bootstrap --force

npm i bootstrap --force (i means install)

**Code currect format**

>alt+shit+f