1. install latest node js
2. npm install @angular/cli
3. ng new (name) –style=scss
4. ng serve
5. npm install [bootstrap@4.0.0-alpha.6](mailto:bootstrap@4.0.0-alpha.6)
6. add .angular-cli.json above styles.scss "../node\_modules/bootstrap/dist/css/bootstrap.min.css",
7. ng serve – open
8. app.component.ts & main.ts, component selector “app-root” is importing data from the index.html file.
9. Define selection in the component file.

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

@Component({

selector:'app-testpage',

templateUrl: './testpage.component.html'

})

export class TestpageComponent{

}

1. Need to addcomponent in the app.module.ts

import {TestpageComponent} from './testpage/testpage.component';

also require to declare like below

@NgModule({

declarations: [

AppComponent,

TestpageComponent

],

1. Easy way to create component with below command. It will automatically create component & also add & declare it in the appmodule.

Ng g c component name

1. 3 types of define selectors attribute(selector:'[app-testpage]’), class(selector:'.app-testpage’) & element(selector:'app-testpage’)

3 type of selector for componants

1. attribute <app-test></app-test> - selector ='app-test'

2. class <div class ="app-test"> - selector ='.app-test'

3. <div app-test> - selector ='[app-test]'

1. String interpolation{{}}, property binding[] & event binding().
2. Angular slider – slick carousel
3. Component to template one way binding, component to template + template to component twoway binding.
4. Interpolation{{text}}: Component to template – ts file
   1. export class InterpolationComponent implements OnInit {
   2. public firstName;
   3. constructor() { }
   4. ngOnInit(): void {
   5. this.firstName = "Interpolation";
   6. }
   7. }
5. Interpolation:html
   1. {{ firstName }}
   2. {{ firstName | uppercase }}
6. Propertybinding[text]: ts file
   1. export class PropertybindingComponent implements OnInit {
   2. public dynamicimagepath;
   3. public isDisabled;
   4. constructor() { }
   5. ngOnInit(): void {
   6. this.dynamicimagepath = "../../assets/feed2x\_new.png";
   7. this.isDisabled = false;
   8. }
   9. }
7. Propertybinding: html
   1. <img [src]="dynamicimagepath"/>
   2. <input type="text" [disabled]="isDisabled">
8. Propertybind if two words than second capital exe.rowSpan.
9. Classbinding [] = ts file
   1. export class ClassbindingComponent implements OnInit {
   2. public testVariable;
   3. public isChangeColor;
   4. constructor() { }
   5. ngOnInit(): void {
   6. this.testVariable = "changecolor";
   7. this.isChangeColor = false;
   8. }
   9. }
10. Classbinding – html file
    1. <div [class]="testVariable">Class Binding text</div>
    2. <div [class.changecolor]="isChangeColor">Class Binding text</div>
11. Classbinding – css file
    1. .changecolor{
    2. color: #ff0000;
    3. }
12. Classbinding – with ngClass
13. tsfile
    1. this.isChangeFontWeight = true;
14. html file
    1. <div [ngClass]='{"changefontweight": isChangeFontWeight, "changecolor": isChangeColor}'>NG Class Binding text</div>
15. Css file
    1. .changecolor{
    2. color: #ff0000;
    3. }
    4. .changefontweight{
    5. font-weight: bold;
    6. }
16. Event binding () - Template to Component () oneway binding method
    1. Ts file
    2. inputchange() {
    3. console.log("Input value change");
    4. }
17. Html file
    1. <input type="text" (change)="inputchange()">
18. Twoway binding[()] = propertybinding[] + event()
19. Ts file

export class TwowaybindingComponent implements OnInit {

  public twowayBinding;

  constructor() { }

  ngOnInit(): void {

  }

}

1. Htmlfile
   1. <input type="text" [(ngModel)]="twowayBinding">
2. Also require to import form in app module.
3. Directives –html files
   1. <h1>directive-example ngif works!</h1>
   2. <div \*ngIf="true">
   3. Conditional show the content
   4. </div>
   5. <button (click)="isVariable=!isVariable">Toggle</button>
   6. <div \*ngIf="isVariable">
   7. Conditional show the content2
   8. </div>
4. Directive – ts file
   1. export class DirectiveExampleComponent implements OnInit {
   2. isVariable = true;
   3. constructor() { }
   4. ngOnInit(): void {
   5. }
   6. }
5. Directive NG switch – HTML files
   1. <h1>directive-example ngswitch works!</h1>
   2. {{currentDay}}
   3. <div [ngSwitch]="currentDay">
   4. <div \*ngSwitchCase="'0'">Sunday</div>
   5. <div \*ngSwitchCase="'1'">Monday</div>
   6. <div \*ngSwitchCase="'2'">Tuseday</div>
   7. <div \*ngSwitchCase="'3'">Wednesday</div>
   8. <div \*ngSwitchCase="'4'">Thursday</div>
   9. <div \*ngSwitchCase="'5'">Friday</div>
   10. <div \*ngSwitchCase="'6'">Saturday</div>
   11. </div>
6. Direct NG swtich TS files
7. export class DirectiveExampleComponent implements OnInit {
8. isVariable = true;
9. constructor() { }
10. public currentDay;
11. ngOnInit(): void {
12. this.currentDay = new Date().getDay();
13. }
14. }
15. Directive NG for – HTML files
    1. <h1>directive-example ngFor works!</h1>
    2. {{days}}
    3. <div \*ngFor="let day of days">
    4. <ul>
    5. <li>{{day}}</li>
    6. </ul>
    8. </div>
16. Directive NG for – TS files 5
    1. public days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
17. Directive ngswitch + ngFor html file
    1. <h1>directive-example ngswitch + ngFor works!</h1>
    2. <div [ngSwitch]="currentDay">
    3. <div \*ngFor="let testday of testdays">
    4. <ul>
    5. <li \*ngSwitchCase="testday.dayNumber">{{testday.dayName}}</li>
    6. </ul>
    7. </div>
    8. </div>
18. Directive ngswitch + ngFor ts file
    1. public testdays = [
    2. { dayNumber: 0, dayName:'Sunday'},
    3. { dayName:'Monday', dayNumber:1},
    4. { dayName:'Tuesday', dayNumber:2},
    5. { dayName:'Wednseday', dayNumber:3},
    6. { dayName:'Thursday', dayNumber:4},
    7. { dayName:'Friday', dayNumber:5},
    8. { dayName:'Saturday', dayNumber:6},
    9. ];
19. Services Method
20. Html files
    1. <ul>
    2. <li \*ngFor="let employee of employees">
    3. {{employee.employee\_name}}
    4. </li>
    5. </ul>
21. Ts files
    1. import { Component, OnInit } from '@angular/core';
    2. import { EmployeeDataService } from '../employee-data.service';
    3. @Component({
    4. selector: 'app-employee-listing',
    5. templateUrl: './employee-listing.component.html',
    6. styleUrls: ['./employee-listing.component.css']
    7. })
    8. export class EmployeeListingComponent implements OnInit {
    9. public employees = []
    10. constructor(private employeeService: EmployeeDataService) { }
    11. ngOnInit(): void {
    12. this.employees = this.employeeService.getEmployees();
    13. }
    14. }
22. Create service using ng g s servicename
    1. export class EmployeeDataService {
    2. constructor() { }
    3. getEmployees() {
    4. return [
    5. { id: "1", employee\_name: "Gopal Joshi", age: 37, gender: "Male" },
    6. { id: "2", employee\_name: "Samik Maniar", age: 38, gender: "Male" },
    7. { id: "3", employee\_name: "Vimal Chauhan", age: 35, gender: "Male" },
    8. { id: "4", employee\_name: "Vivek Joshi", age: 22, gender: "Male" },
    9. { id: "5", employee\_name: "Rashid Shaikh", age: 32, gender: "Male" },
    10. ]
    11. }
    12. }
23. 5
    1. public testdays = [
24. App Module file – “Component – declaration”, “Module – Imports”, “Services - Providers6”
25. 5
26. 1
27. 2
28. 5
29. 5
30. 6