Angular

1) What is Angular?

- Angular is a TypeScript-based open source framework for building client-side web applications.
- Angular is a platform and framework for building single-page applications using HTML and TypeScript.
- Angular is written in TypeScript.

2) What are the Angular features?

- Angular supports Mobile, Tablets etc.
- Speed and Performance.
- Modular.
- ES6 and TypeScript support.

3) What is difference between Angular and AngularJS?

Angular	AngularJS
Angular is a modern web Application framework build entirely in TypeScript- Its superset of Java	AngularJS is a frontend MVC framework based on JavaScript programming language.
Angular utilizes services/controller architecture.	AngularJS follows the MVC architecture.

4) What is decorator in Angular?

- Decorator provides metadata means data about data and the notation is @
- That determines how the component should be proceed, instantiated, and used at runtime.
- e.g @Component

5) Which Compiler have used in Angular?

- JIT

6) How to compile angular code?

- Using ng serve command we can run angular code.

7) If one port is already running Can we Run ng serve command in other terminal

- **Yes,** We can run but we need to use another terminal and follow this command.
- ng serve --port 4205 (u can give another port number also).

8) What is component ?

- Component is a basic building block of our angular application.
- Component is decorator which will provide metadata to particular class that component how that should be initiated proceed at the run time.
- It have its own business logic (app.component.ts)
- It have its own template (app.component.html)
- It have its own styling (app.component.css)
- It have its own TypeScript (app.component.spec.ts)
- Components have 4 Files are
  - app.component.ts
  - app.component.html
  - app.component.css
  - app.component.spec.ts

9) How can we create a component?

- There are two ways we can create component 1) Manually 2) Using angular CLI
- Using **Manually** we need to create folder and **app module (4 files).**
- 4 files are
  - app.component. css
  - app.component.html
  - app.component.spec.ts
  - app.component.ts
    - We need to create @Component directive({
    - selector: 'app-root', → is a reference of app component
    - templateUrl: './app.component.html', → reference of html file
    - styleUrls: ['./app.component.css'] }) → reference of css file
    - Also need to create class AppComponent

- Using **Angular CLI** we just need to type a command ng g c componentName
- ng stands for → Angular
- g stands for → Generate
- c stands for → Component
- componentName → U can give any component Name.

10) How many ways we can create a template ?

- There are two ways to create template in components.
- There is no major difference in both inline and external.
- Inline and External:
- But there is one difference.
- **Inline :** When we have small code that time we can use **Inline Template.**
- **External:**When we have complex code that time we can use **External Template.**

11) What is the syntax to create component in Angular?

- Using **Angular CLI** we just need to type a command ng g c componentName
- ng stands for → Angular
- g stands for → Generate
- c stands for → Component
- componentName → U can give any component Name.

12) If we created Component then where we need to declare that Component?

- Inside app.module.ts →@NgModule → declaration [componentName].

13) What is Data Binding?

- Data binding means communication between Component and View.
- Component stands for TypeScript file.
- View stands for template file (HTML file).

14) How many type of Data Binding?

- There are two types of Data Binding.
- **One way data binding :**
  - One way data binding means data flow in one direction.
  - When we want to bind our data from TypeScript file to View/ view file to TypeScript file that time we can use one way data binding.
  - HTML file $\longrightarrow$ TypeScript file
  - HTML file $\longleftarrow$ TypeScript file
- **Two way data binding :**
  - Two way data binding means data flow in both direction.
  - When we want to bind our data from TypeScript file to View & view file to TypeScript file that time we can use two way data binding.
  - Html file $\longleftrightarrow$ TypeScript file
  - **Syntax is : [] + () = [(ngModel)]**

15) What is ngModule?

- ngModule is a Directive.
- It's a combination of property binding and event binding also known as **Banana in Box.**
- To achieve the two way data binding angular uses ngModel – directive.
- It bind like input, select, selectArea.
- ngModel is not a part of angular core library.

- It is a part of **FormsModule** library.
- FormsModule is present in **AngularForms**.

15) How many types of one way data binding?

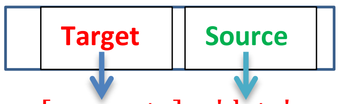
- String Interpolation :
  - Syntax of String Interpolation is {{data}}
  - Means data flow from TypeScript to HTML.
  - If we want to show expression then uses String Interpolation.
  - The content inside the double curly braces {{content}} known as **Template Expression.**
  - The Angular first evaluates the Template Expression and converts it into a string.
  - Then it replaces Template Expression with the result in the original string in the HTML.
  - Whenever the template expression changes, the angular updates the original string again..
  - {{templateExpression}}.
  - E.g.
  - In Component define firstname and lastname as string values .

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  firstName= 'Sachin';
  lastName="Tendulkar"
}


```

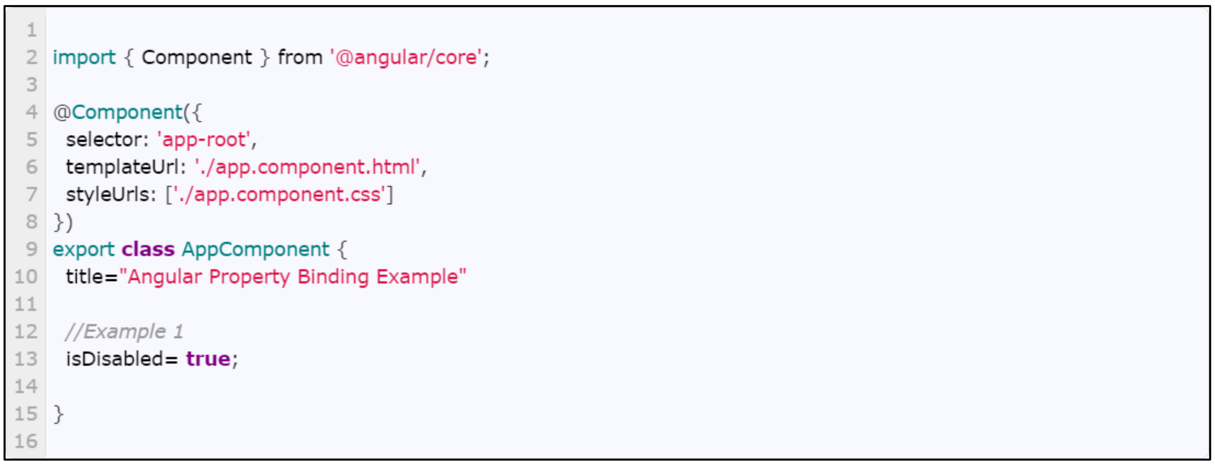
  - The Angular replaces both {{firstName}} & {{lastName}} with the values of firstName & lastName variable from the component.

```

Welcome,  {{firstName}} {{lastName}}

```

  - If your run the app you will see **Welcome, Sachin Tendulkar** on window.
  - Whenever we change the values of firstName and lastName, Angular updates those values in HTML template.

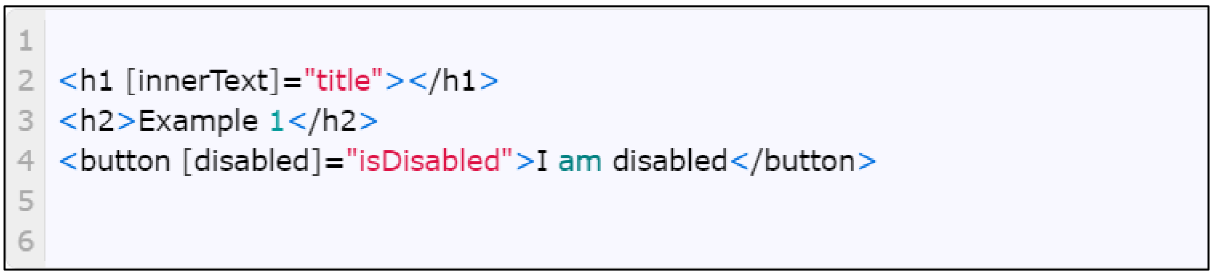


- Property Binding :
  - Syntax of Property Binding : [property] = 'data'
  - Here [property] is a target and 'data' is a source.
  - Property binding allows us to bind the component property to a value in the template.
  - Any changes to the property will reflect in template.
  - Data flow from TypeScript to HTML.
  - If we want to assign any dynamic attribute then we can use it
  - e.g img, button..
  - We can also use it to set the properties of custom components or directives (property decorator with @Input).
  - E.g. declare title = "Angular Property Binding Example".
  - And isDisabled = true. Now Bind this values in HTML template.
  - App.component.html



```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title="Angular Property Binding Example"

  //Example 1
  isDisabled= true;

}

```

  - App.component.ts



```

<h1 [innerText]="title"></h1>
<h2>Example 1</h2>
<button [disabled]="isDisabled">I am disabled</button>


```

  - The title property of the component class is bound to the innerText property of the h1 tag.
  - Disabled Property of the button is bound to the isDisabled Property of the component.
  - Whenever we modify the title or isDisabled in the component, the Angular automatically updates the HTML Template.

- Event Binding:
  - Syntax of Event Binding is : (target-event) = 'TemplateStatement".
  - When we want to send data from HTML to TypeScript then we can use Event Binding.
  - Event binding allow us to bind the DOM event to a method in a component the event trigger the corresponding method is executed.
  - Event binding allows us to bind events such as keystrokes, clicks, hover, touch, etc to a method in component.
  - It is one way from view to component.

16) How many types of two way data binding?

- There is only one way using ngModule directive.
- It's a combination of property binding and event binding
- **Syntax is : [] + () = [(ngModel)].**
- Angular have a special directive ngModel, which sets up the two-way binding.

17) How can we pass data from one component to Another Component?

- We use Property Decorator. If there is relation between Both Components.

18) What is Property Decorator?

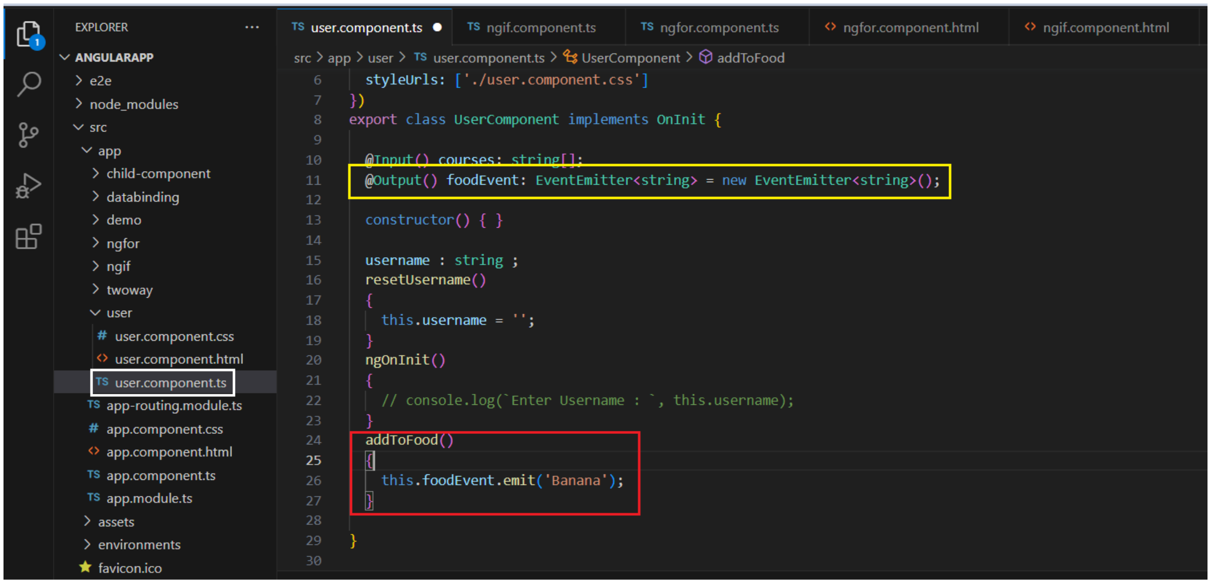
- If we want to pass the data from one component to another component then we can use Property Decorator.
- There are two types of property decorator.
- **@Input :** If we want to **send data** from parent component to child component
- **@Output :** If we want to **send data** from child component to parent component.
- **Always** remembered we need to **declare** both properties in **child component**.

19) What is EventEmmiter ?

- EventEmmiter is a class and it is used for creating custom event.
- In Angular have two types of events System Events & Custom Events.
- System events : input, click, keyup, keydown, scroll
- Custom event means we can create our own event.

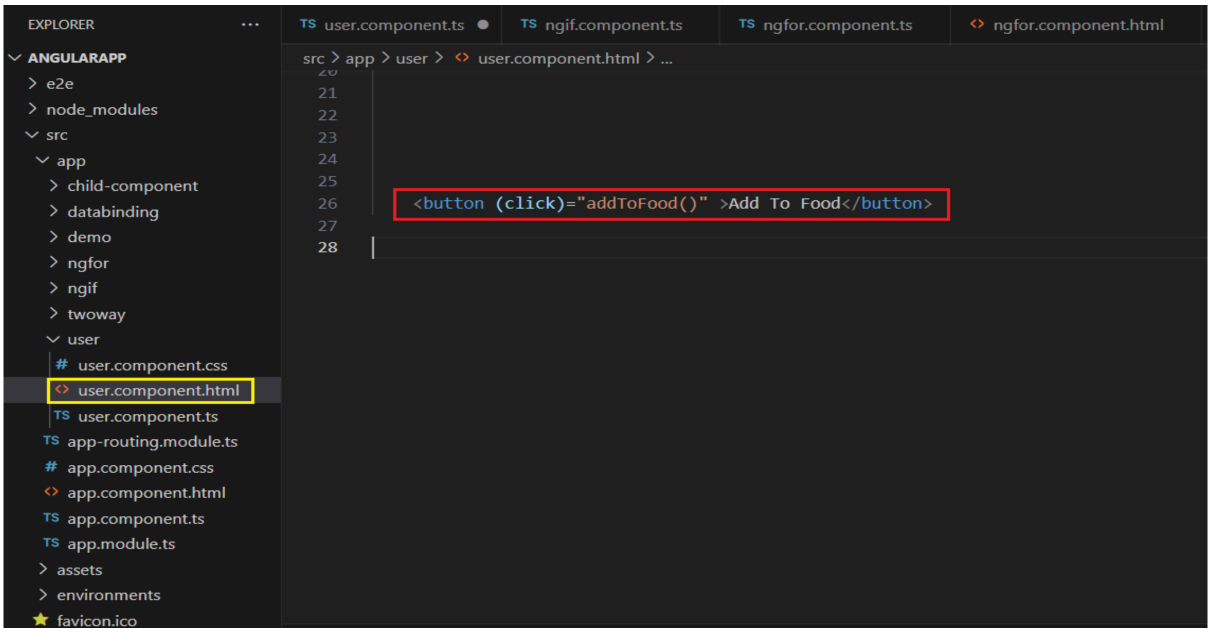
20) How to create custom event ?

- First we need to define the Event using **@output ()** decorator with **EventEmmiter** class.
- Let's make **foodEvent ()** in child component and emit **Banana** in addToFood ().



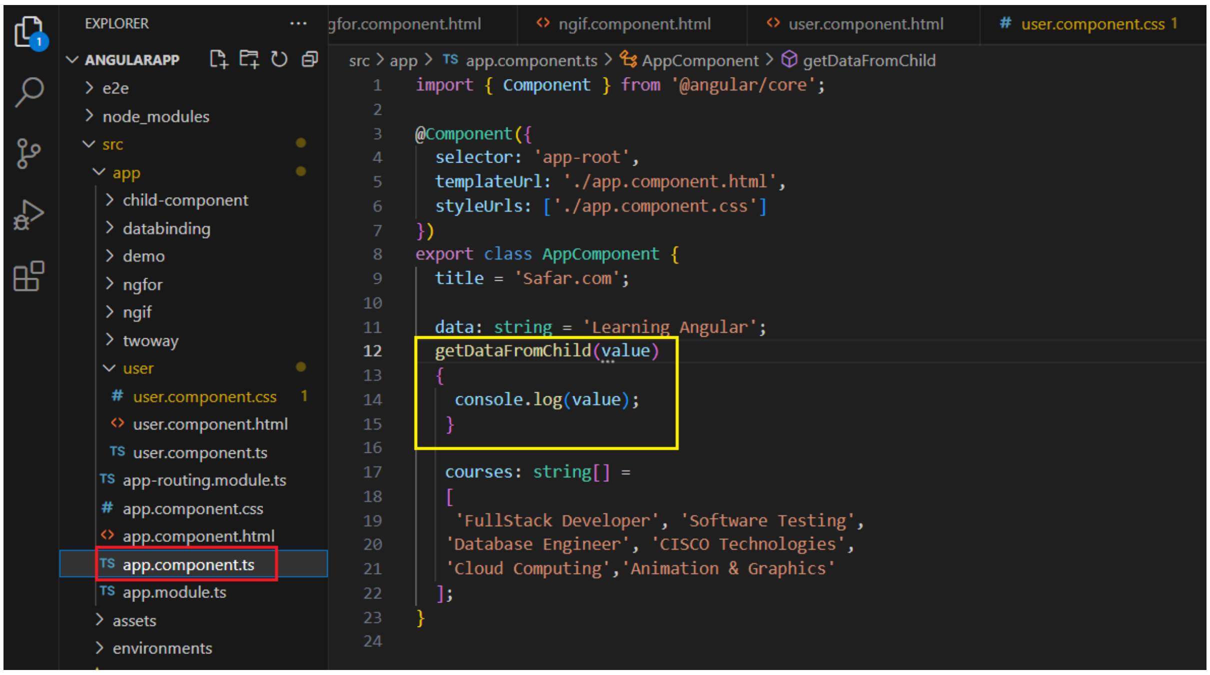
```
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {

  @Input() courses: string[];
  @Output() foodEvent: EventEmitter<string> = new EventEmitter<string>();

  constructor() { }

  username : string ;
  resetUsername()
  {
    this.username = '';
  }
  ngOnInit()
  {
    // console.log(`Enter Username : `, this.username);
  }
  addToFood()
  {
    this.foodEvent.emit('Banana');
  }

}
```

- Then create a button and perform click event on **addToFood().**
- Here your own event is created.



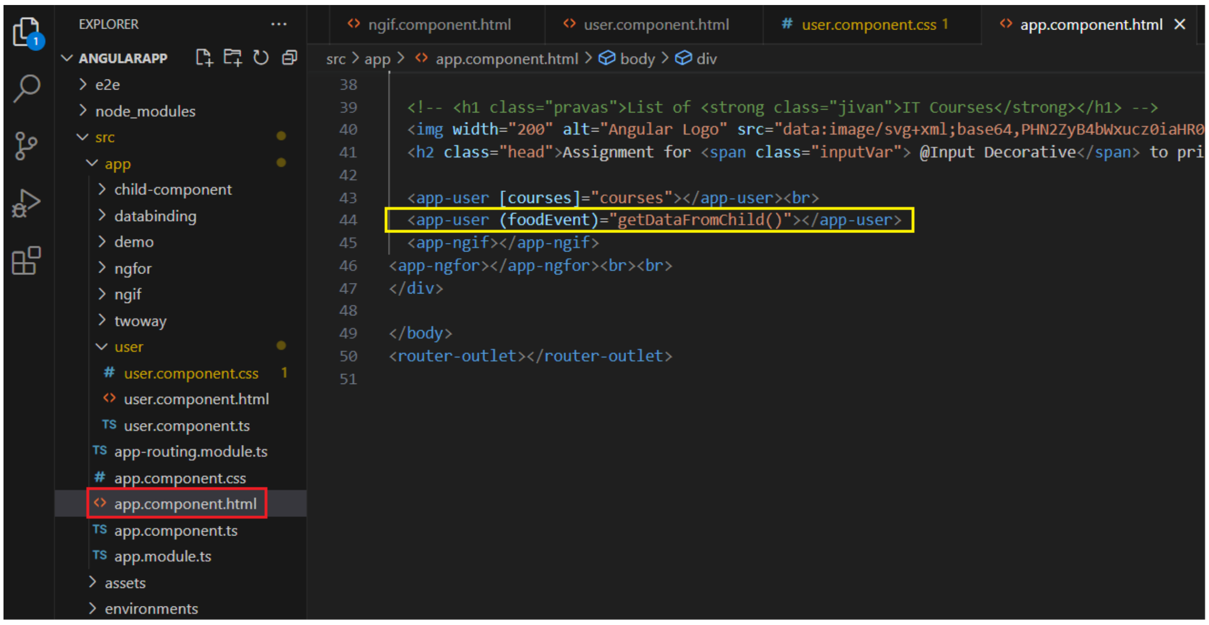
```
<button (click)="addToFood()" >Add To Food</button>
```

- Now let's move on Parent Component.



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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Safar.com';

  data: string = 'Learning Angular';
  getDataFromChild(value)
  {
   console.log(value);
  }

   courses: string[] =
   [
    'FullStack Developer', 'Software Testing',
   'Database Engineer', 'CISCO Technologies',
   'Cloud Computing','Animation & Graphics'
   ];
}
```

- Then bind the data and print value Banana value on console.



```

  <!-- <h1 class="pravas">List of <strong class="jivan">IT Courses</strong></h1> -->
  <img width="200" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0
  <h2 class="head">Assignment for <span class="inputVar"> @Input Decorative</span> to pri

  <app-user [courses]="courses"></app-user><br>
  <app-user (foodEvent)="getDataFromChild()"></app-user>
  <app-ngif></app-ngif>
<app-ngfor></app-ngfor><br><br>
</div>

</body>
<router-outlet></router-outlet>
```

21) What is @event ?

- Angular includes $event that contains the information about an event.

22) What is Directive?

- By using Directive we can modify the DOM elements and change behavior of DOM element.
- Basically directives used to manipulate the DOM element.

23) How many directives we have?

- Angular have 4 directives.
- Component Directive
- Structural Directive
- Attribute Directive
- Custom Directive

24) What is Component directive ?

- Component directive are used in main class.
- They contain the details of how the component should be processed, instantiated and used at runtime.

25) What is Structural directive?

- Structural directive starts with * sign.
- These directives are used to manipulate and change the structure of the DOM element. And structural directive have 3 types :
- ***ngIf** : The *ngIf allow us to Add/Remove the DOM element.
- ***ngSwitch** : Iterate based on the condition. Add/Remove the DOM element..
  - [ngSwitch] : is Angular directive allow us to display one or more DOM element.
  - *ngSwitchCase : match the expression.
- ***ngFor** → *ngFor directive is used to repeat the portion of HTML Template .

26) What is Attribute directive?

- ✓ Attribute directives are used to change the look and behavior of the DOM Element.
- ✓ Having 2 types :
- ✓ **ngStyle :**
  - Used for setting the styles of DOM Elements.
  - We can set one or more properties also we can pass the dynamic value.
  - E.g. <div [ngStyle]="{'color': colorVal, 'background-color': '#ddd'}">
  - We have apply color on text and background-color also.
  - We can apply more than value
  - Means we can apply only one time on one element.
  - The ngClass directive is used to add or remove CSS classes.
  - main use: when use dynamic value

- ✓ **ngClass :**
  - The ngClass directive allows us to set the css class dynamically for the DOM Element.