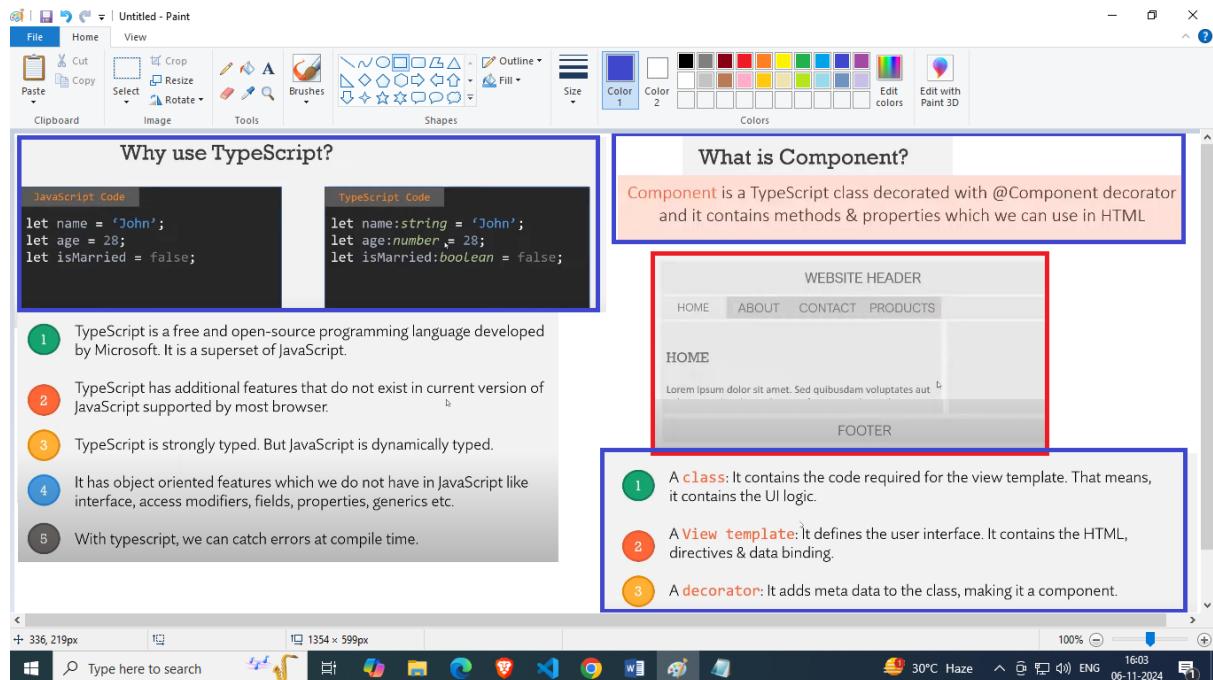


The diagram illustrates the structure of an Angular application. At the top, a browser window icon is shown next to the title "Working with Angular Project | Angular Basics | Angular 12+". Below it, the title "Running Angular Application" is displayed. Three main components are shown: 1) A code snippet for `index.html` which contains the HTML structure of a webpage, including a title and an `<app-root>` tag. 2) A code snippet for `app.component.ts` which defines an Angular component with a selector of `'app-root'`. 3) A code editor window showing the content of `app.component.html`, which contains an `<h2>` tag with a placeholder {{title}} and a `<p>` tag with the text "This is my first Angular App". Arrows indicate the relationship between these files: the browser icon points to the `index.html` code, and the other two code snippets point to the `app.component.html` code.

The video segment is titled "Working with Angular Project | Angular Basics | Angular 12+" and "Why use Angular?". It features a list of three reasons:

- 1 When we run `angular` app, `index.html` file gets rendered in the webpage. This is the main HTML file whose content will change when we navigate around or do some other stuff on webpage.
- 2 An angular app consists of component and by default angular CLI provides us with one App Component.
- 3 Each component has four important files `component.ts`, `component.html`, `component.css` & `component.spec.ts` file. `component.ts` is the main component file.



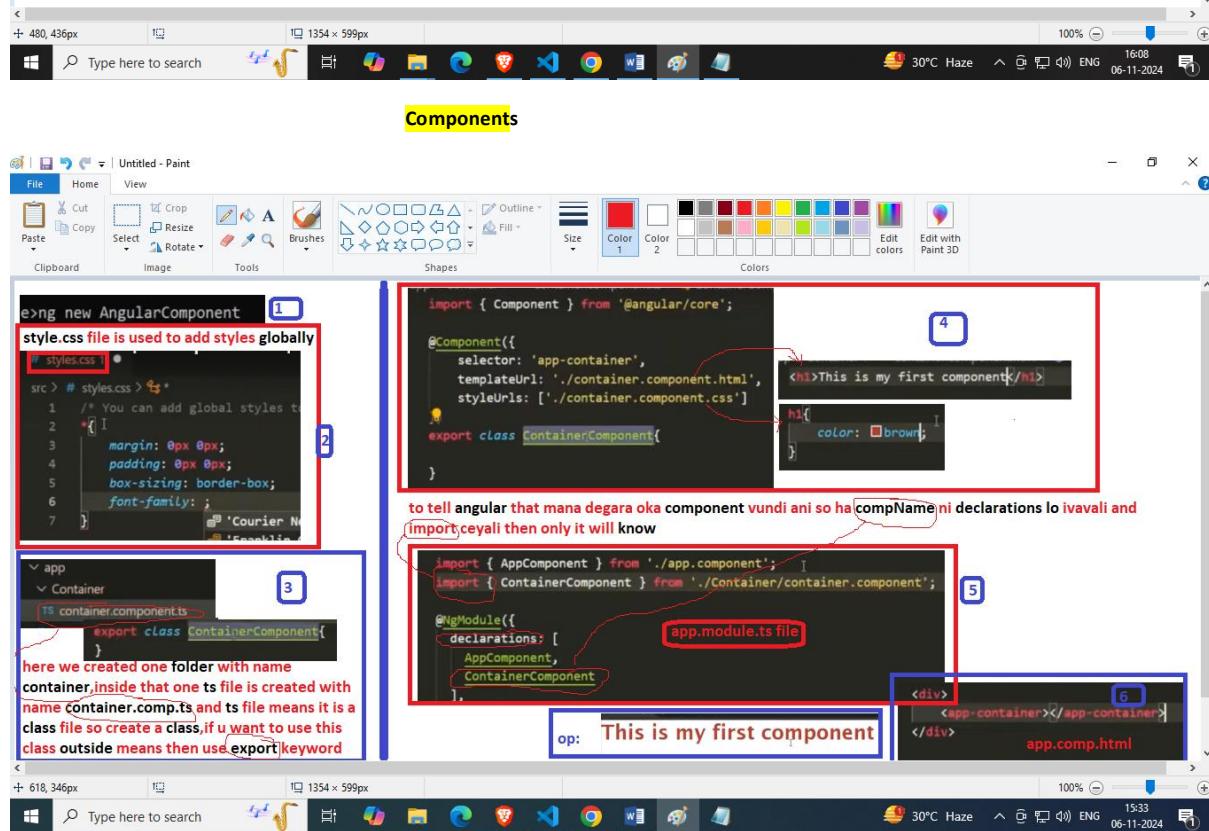
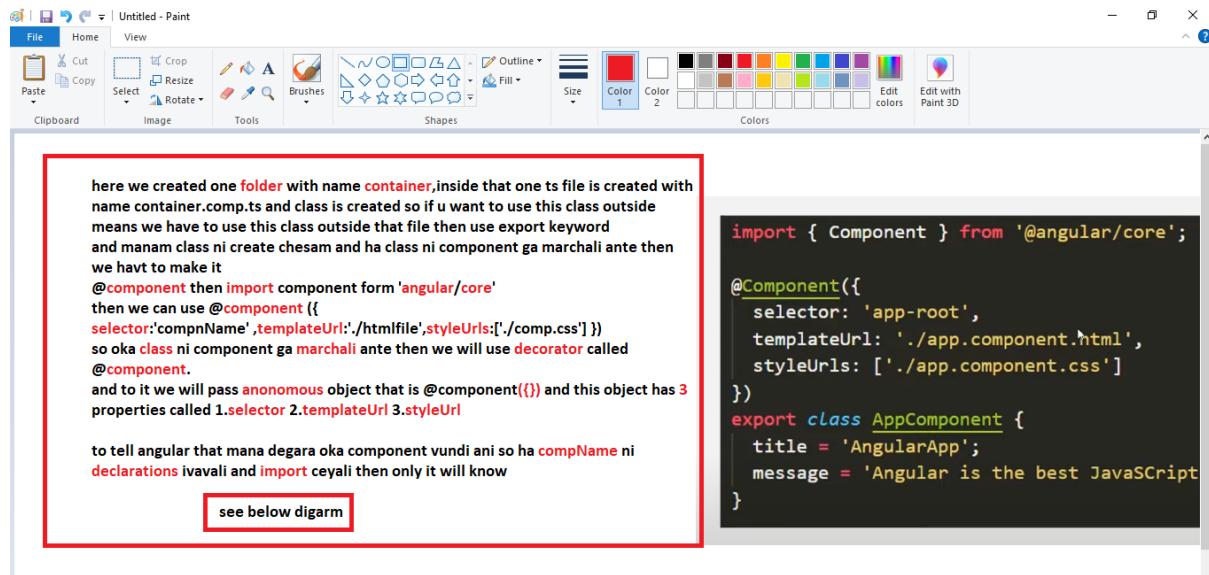
here we created one folder with name container, inside that one ts file is created with name container.comp.ts and class is created so if u want to use this class outside means we have to use this class outside that file then use export keyword

and manam class ni create chesam and ha class ni component ga marchali ante then we havt to make it @component then import component form 'angular/core'

then we can use @component ({ selector:'compnName' ,templateUrl:'./htmlfile',styleUrls:['./comp.css'] })

so oka class ni component ga marchali ante then we will use decorator called @component and to it we will pass anomous object that is @component({}) and this object has 3 properties called 1.selector 2.templateUrl 3.styleUrl

to tell angular that mana degara oka component vundi ani so ha compName ni declarations ivavali and import ceyali then only it will know



Data Binding

The screenshot shows a Microsoft Paint window titled "Untitled - Paint" containing code examples for Angular Data Binding.

Component Class:

```
component class
export class MyComponent{
  title = 'eShopping';
  slogan = 'Your one stop shop';
  display = false;

  onClick(){
    this.display = true;
  }
}
```

View Template:

```
<div class="header">
<div>{{ title }}</div>
<div>{{ slogan }}</div>
<button (click)="onClick()"></button>
<div [hidden]="display">
<p>Display this content</p>
</div>
</div>
```

Code Snippet (TS file):

```
ngOnInit(): void {
}
slogan: string = 'Your one stop shop for everything';
getSlogan(){
  return 'This is a new slogan for this site';
}

<div class="site-slogan">
<h2>{{ slogan }}</h2>
<h2>{{ 12 + 35 + 78 | }}</h2>
<h2>{{ getSlogan() }}</h2>

```

Annotations:

- A red box highlights the "String interpolation" section in the component class code.
- A red box highlights the "arithmatic operations function" section in the TS file code.
- A diagram at the bottom shows "Data flow from component to class view template" with an arrow labeled "String interpolation: {{data}}".
- A callout box states: "string interpolation: it is used to bind data from TS file to View(HTML) file. It performs operations only for strings".

The screenshot shows a web browser displaying a website titled "eShopping".

Header:

AngularComponent ngClass Directive in Angular | Directives | Angular 12+

Navigation:

HOME ABOUT CONTACT CART PRODUCTS

Hero Section:

Your one stop shop for everything.

Search Bar:

Search: sgsdg You searched for "sgsdg"

Product Listings:

- Minimalists Analog Watch
Available Price:\$109 Color:Black Show Details Buy Now
- Hisense Ultra HD Smart TV Show Details Buy Now

Directives

What is Directive?

Directives are simply an instruction to the DOM

Types of Directive?

- 1 **Structural directive:** Changes the view of a webpage by adding or removing DOM elements from a webpage.
- 2 **Attribute Directive:** Used like an attribute on a existing webpage element to change its look and behaviour.

ngFor Directive?

- 1 The `ngFor` directive is used to repeat a portion of HTML template once per each item from an iterable list.
- 2 The `ngFor` is a structural directive. That means, `ngFor` manipulates the DOM by adding or removing elements from the DOM.

ngFor

```
<div *ngFor="let item of [1, 3, 5]">
  <p>Number is: {{ item }}</p>
</div>
```

Number is: 1
Number is: 3
Number is: 5

-----CSS-----

```
.product-container{ display: flex; margin: 15px 0px; padding: 5px 5px; border-radius: 5px; }
.image-container{ background-color: #fff; height: 70px; width: 70px; padding: 10px 0px; text-align: center; }
.detailContainer{ display: flex; margin: 0px 20px; }
.nameContainer{ margin: 0px 20px; }
.details{ width: 140px; }
```

adding css below dig

```
.name-container{
  margin: 0px 20px
}
.details{
  width: 140px;
}
.options{
  width: 220px;
  margin-left: 80px;
}
.btn{
  border-radius: none;
  margin: 0px 5px;
  padding: 2px 10px;
  font-size: 12px;
}
```

+ 507, 520px 1354 x 599px 100% 31°C Haze 06-11-2024

products.component.ts

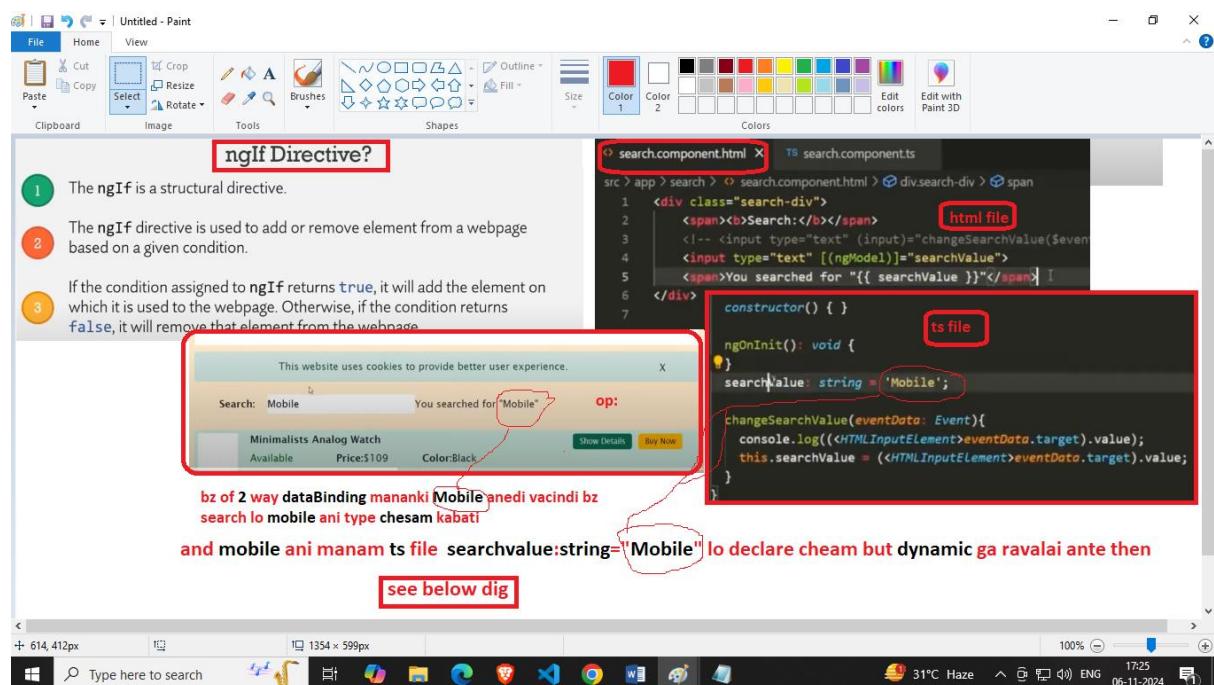
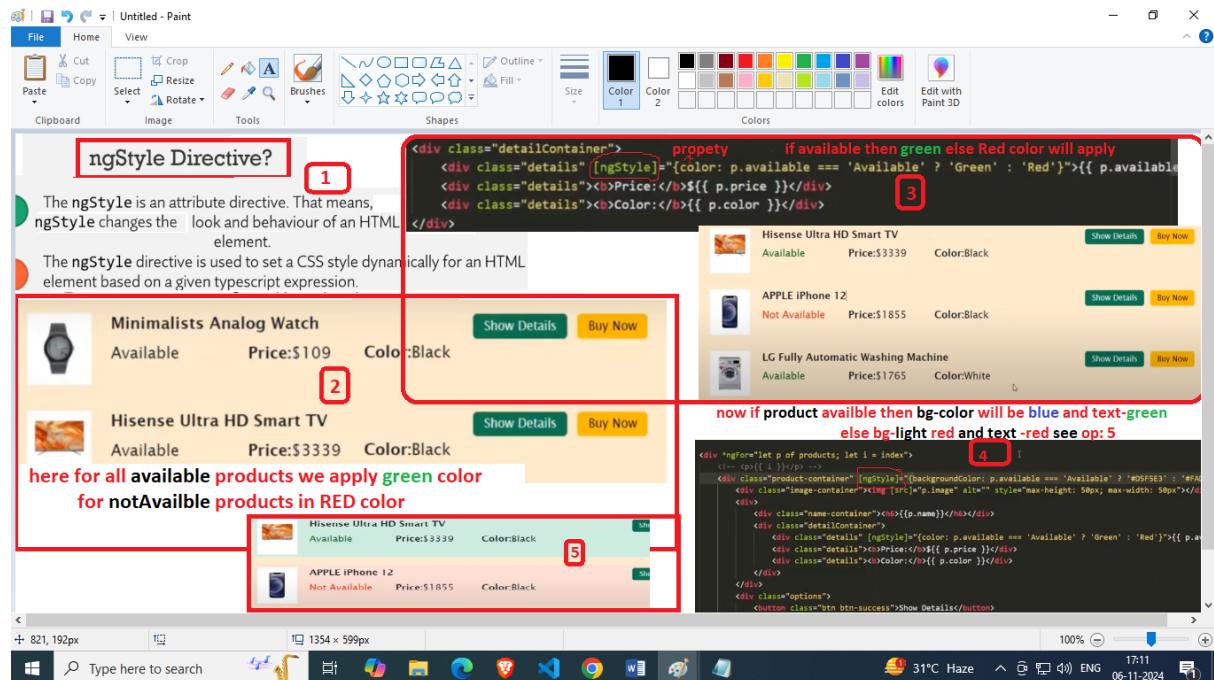
```
1 import { Component, OnInit } from '@angular/core'
2
3 @Component({
4   selector: 'app-products',
5   templateUrl: './products.component.html',
6   styleUrls: ['./products.component.css']
7 })
8 export class ProductsComponent implements OnInit
9
10 constructor() { }
11
12 ngOnInit(): void {
13 }
14 products = [
15   {id:1, name: 'Minimalists Analog Watch', price: '109', color: 'Black', available: 'Not Available', image: '/assets/products/product-image-3.jpeg'},
16   {id:2, name: 'Hisense Ultra HD Smart TV', price: '3399', color: 'Black', available: 'Available', image: '/assets/products/product-image-4.jpeg'},
17   {id:3, name: 'APPLE iPhone 12', price: '1855', color: 'Black', available: 'Not Available', image: '/assets/products/product-image-5.jpeg'},
18   {id:4, name: 'LG Fully Automatic Washing Machine', price: '1765', color: 'White', available: 'Available', image: '/assets/products/product-image-6.jpeg'},
19   {id:5, name: 'LG Refrigerator with Door Cooling', price: '2815', color: 'White', available: 'Not Available', image: '/assets/products/product-image-7.jpeg'},
20   {id:6, name: 'DELL Inspiron One 7 Ryzen 7', price: '2145', color: 'White', available: 'Available', image: '/assets/products/product-image-8.jpeg'}
21 ];
22 }
```

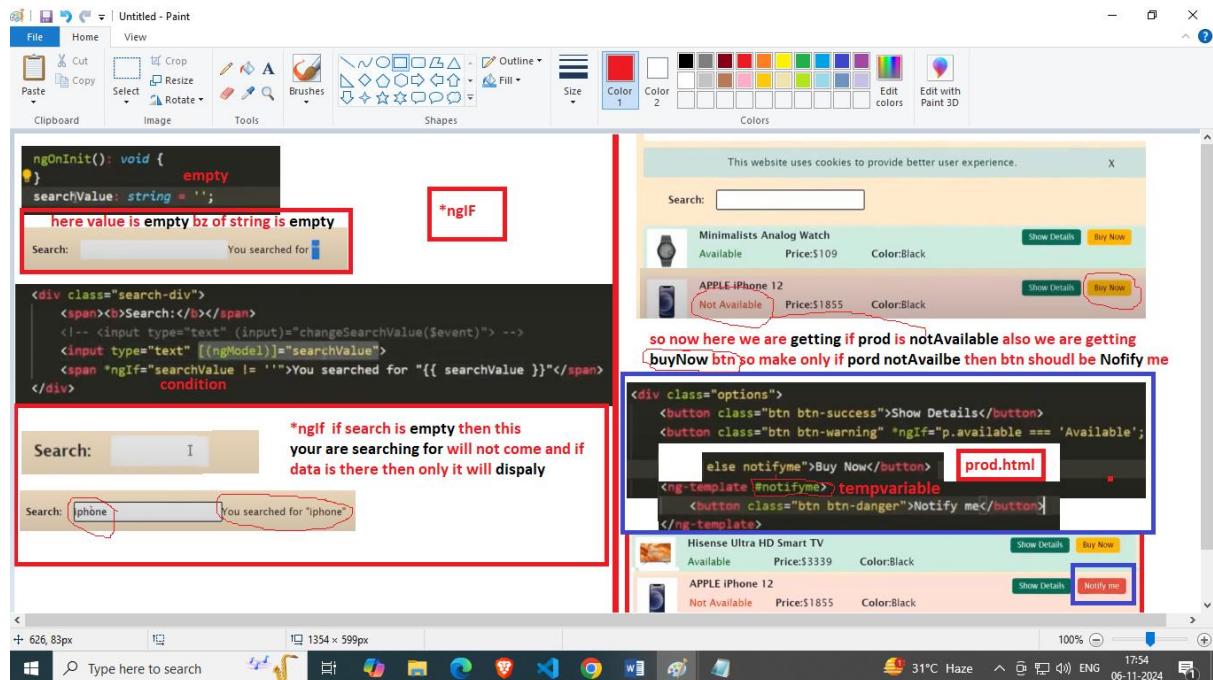
ts file

2

html file

+ 622, 116px 1354 x 599px 100% 31°C Haze 06-11-2024





angular tasks for beginners - YouTube

What is Child Component | Components | Angular 12+

Custom event binding?

1 We can pass data from component class to view template and vice versa using property binding, string interpolation and event binding.

2 We can also pass data from parent component to child component and vice versa. We use @Input & @Output decorator for that.

1.47 / 11:46

Complete Angular 13 Course Step by... procademy - 23 / 105

- ngIf Directive in Angular | Directives | Angular 12+ procademy
- ngClass Directive in Angular | Directives | Angular 12+ procademy
- What is Child Component | Components | Angular 12+ procademy
- Custom Property Binding: @Input Decorator | Data Bindin... procademy
- Custom event Binding: @Output Decorator | Data Bindin... procademy
- Implementing Search

ANGULAR 13+ CHAMBERS COMING SOON

Custom Property Binding: @Input Decorator | Data Bindin... procademy

27°C Haze 18:14 06-11-2024

```

export class CoursesComponent{
  courses = [
    { id:101, name:'JavaScript for beginners', price: 0.00, ratings: 3.5, image:'assets/js.jpg', description: 'In this course you will learn the fundamentals of JavaScript. This course is...'},
    { id:102, name:'Angular for beginners', aut...
  ]
}

app
  courses
    # courses.compon...
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
...
```

@Input

angular tasks for beginners - YouTube

What is Child Component | Components | Angular 12+

Complete Angular 13 Course Step by...

23 / 105

procademy - 23 / 105

procademy

JavaScript for beginners

Angular for beginners

React for beginners

Custom Property Binding: @Input Decorator | Data Bindin...

Custom event Binding: @Output Decorator | Data...

Implementing Search

Custom Property Binding: @Input Decorator | Data Bindin...

Custom event Binding: @Output Decorator | Data...

Implementing Search

Untitled - Paint

File Home View

Cut Copy Paste Select Crop Resize Rotate Tools Shapes Colors

now we will create search and filter component and this 2 components will be child component of the course component

search comp.ts file

```


<div class="search-container">
    <span>Search:</span>
    <input type="text">
  </div>


```

search comp.css file

```

.search-container{
  margin:10px 0px;
}
input[type=text]{
  width: 240px;
  height: 30px;
  padding: 15px;
  border: none;
  margin-left:10px;
}

```

filter html file

```


<div class="filter-container">
    <span>Filter:</span>
    <input type="radio" name="filter" value="All" />
    <span>All Courses</span>
    <input type="radio" name="filter" value="Available"/>
    <span>Free Courses</span>
    <input type="radio" name="filter" value="Not Available"/>
    <span>Premium Courses</span>
  </div>


```

filterer css file

```

.filter-container{
  margin:10px 20px;
}
input[type=radio]{
  margin-left:10px;
}

```

Angular 12

here manam search comp ni course .html file use cheskuntunam

proacademy

same file but added filter component so in op also both components added search and filter

proacademy

components added search and filter

filter.comp.html

```


<div class="filter-container">
    <span>Filter:</span>
    <input type="radio" name="filter" value="All" />
    <span>All Courses</span>
    <input type="radio" name="filter" value="Available"/>
    <span>Free Courses</span>
    <input type="radio" name="filter" value="Not Available"/>
    <span>Premium Courses</span>
  </div>


```

course.comp.html

```


<div class="course-container" *ngFor="let course of courses">
    <div class="course-card">
      <img [src]="course.image" style="width:220px;" alt="Course thumbnail" />
      <div>
        <h3>{{course.title}}</h3>
        <p>Type: {{course.type}}</p>
        <p>Price: {{course.price}}</p>
        <p>Duration: {{course.duration}}</p>
        <p>Instructor: {{course.instructor}}</p>
        <button>View Details</button>
      </div>
    </div>
  </div>


```

Untitled - Paint

File Home View

Cut Copy Paste Select Crop Resize Rotate Tools Shapes Colors

here static data is coming for filter from ts file to make it dynamic we will use @Input directive

filter.ts file

```

constructor() { }
ngOnInit(): void {
  all: number = 0;
  free: number = 0;
  premium: number = 0;
}

```

filter.html

```


<span>Filter:</span>
  <input type="radio" name="filter" value="All" />
  <span>{{'All Courses('+all+')'}}</span>
  <input type="radio" name="filter" value="Available"/>
  <span>{{'Free Courses('+free+')'}}</span>
  <input type="radio" name="filter" value="Not Available"/>
  <span>{{'Premium Courses('+premium+')'}}</span>


```

course.comp.html

```


<div class="course-container" *ngFor="let course of courses">
    <div class="course-card">
      <app-search></app-search>
      <app-filter></app-filter>
    </div>
  </div>


```

here filterComp is the childComp for course comp so transfer data from parent -child we will use @Input

course.ts file

```

getTotalCourses(){
  return this.courses.length;
}
getTotalFreeCourses(){
  return this.courses.filter(course => course.type === 'Free').length;
}
getTotalPremiumCourses(){
  return this.courses.filter(course => course.type === 'Premium').length;
}

```

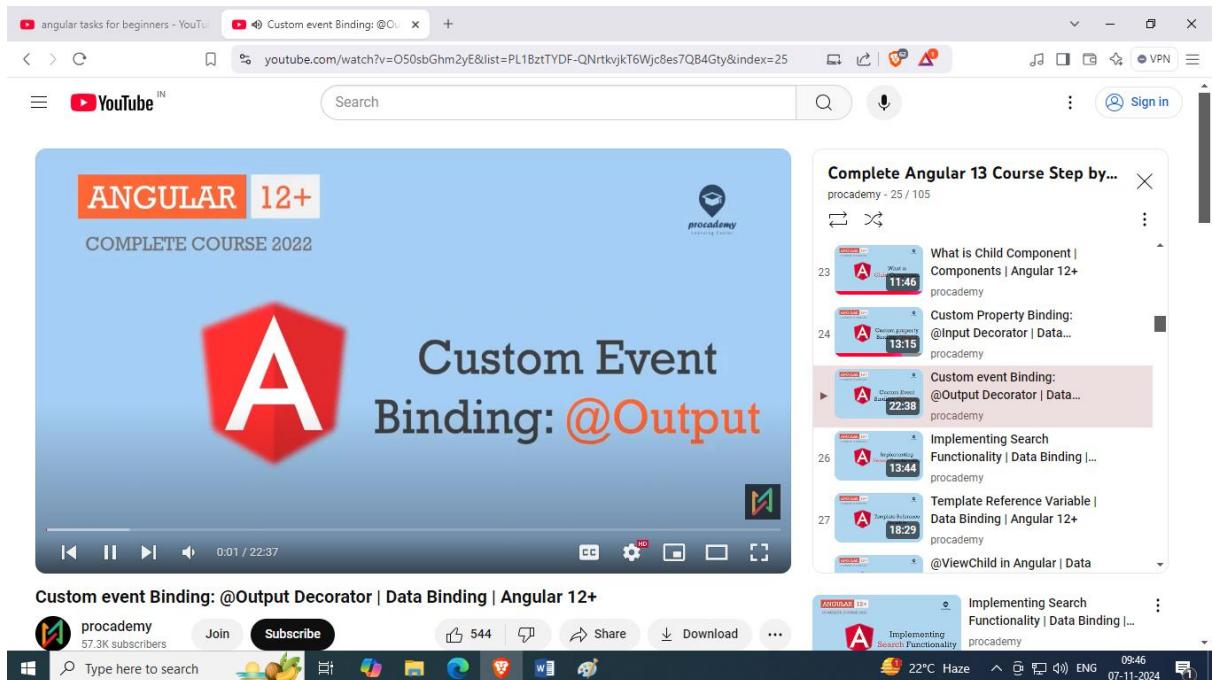
course.comp.html

```


<div class="course-container" *ngFor="let course of courses">
    <div class="course-card">
      <app-search></app-search>
      <app-filter
        [all]="getTotalCourses()"
        [free]="getTotalFreeCourses()"
        [premium]="getTotalPremiumCourses()"
      ></app-filter>
    </div>
  </div>

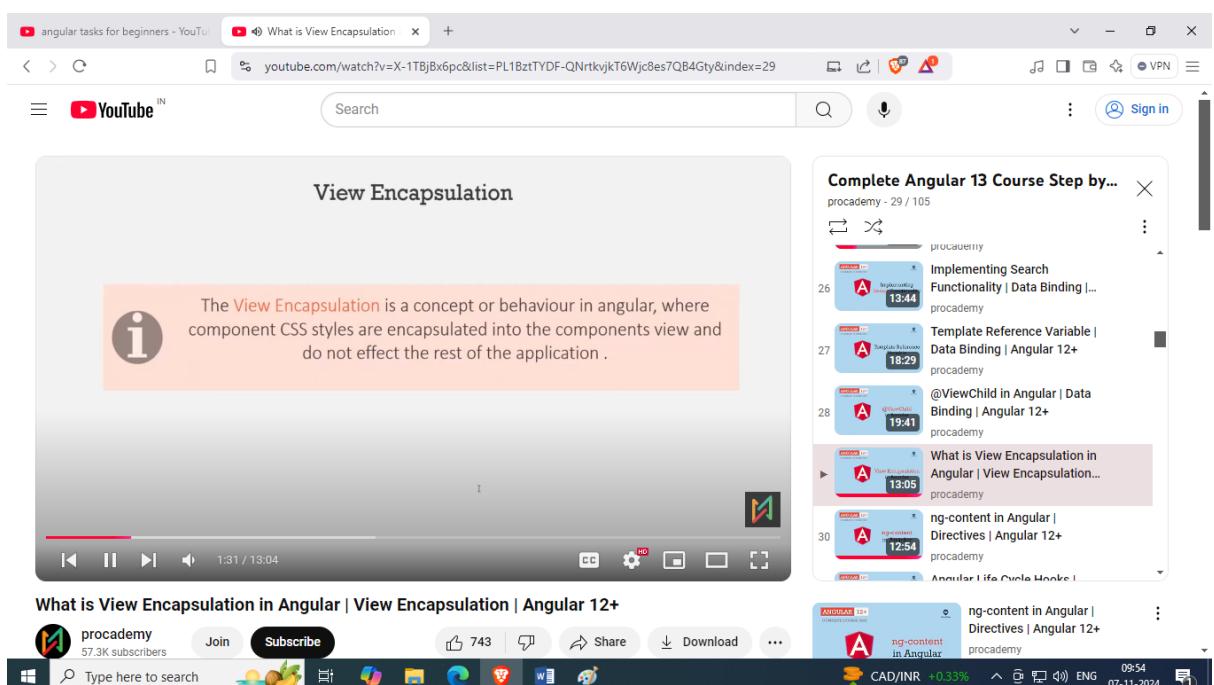

```

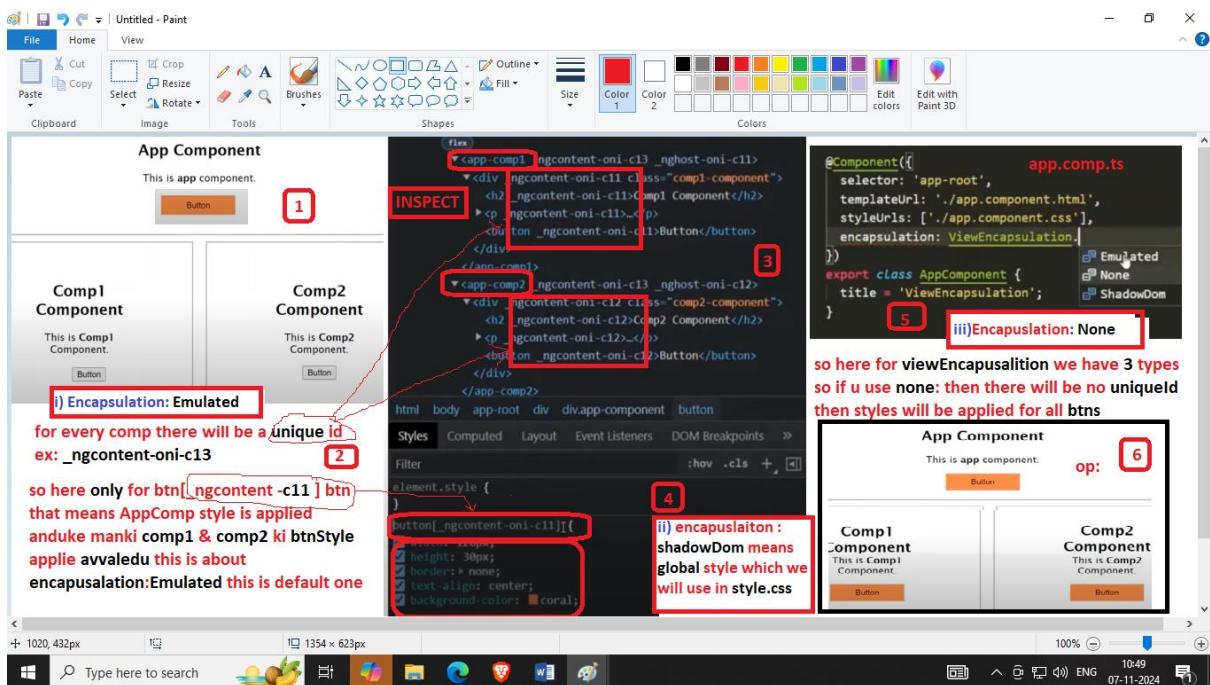
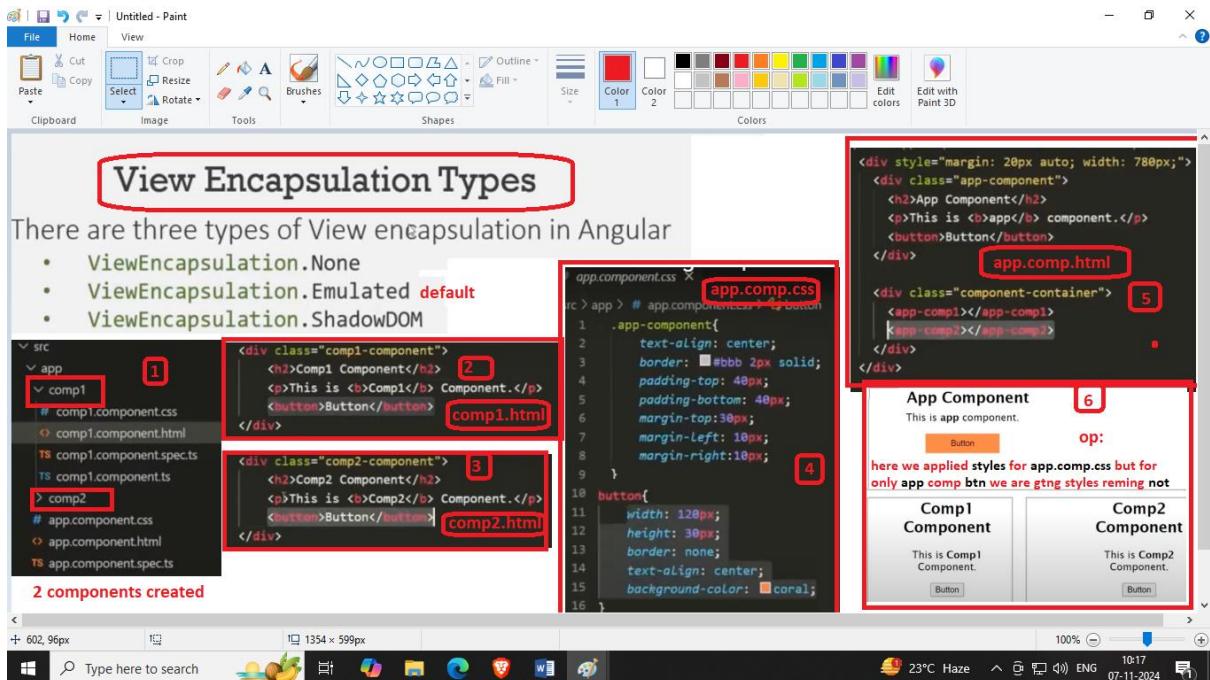
this fields are coming dynamically



4 vdos skiped

View Encapsulation





angular tasks for beginners - YouTube @ContentChild in Angular | +

youtube.com/watch?v=O8UxAgcF-ks&list=PL1BztTYDF-QNrtkvjkT6Wjc8es7QB4Gty&index=33

YouTube IN Search Sign in

ANGULAR 12+
COMPLETE COURSE 2022

@ContentChild in Angular

0:02 / 9:55

Understanding Operators of RxJS | Observables | Angular...
Subjects in RxJS | Observables | Angular 12+
Unsubscribe to an Observable | Observables | Angular 12+
What is Routing in Angular | Angular Router | Angular 13+
Implement Page not found Route | Angular Router |...
Configuring navigation links for Route | Angular Router |...

Custom Attribute Directive | Directives | Angular 12+
procademy

57.3K subscribers Join Subscribe 268 Share Download ...

Type here to search

Angular 12+ COMPLETE COURSE 2022

Custom Attribute Directive | Directives | Angular 12+ procademy

10:51 07-11-2024

angular tasks for beginners - YouTube Custom Attribute Directive | +

youtube.com/watch?v=DxfAEhdOI_0&list=PL1BztTYDF-QNrtkvjkT6Wjc8es7QB4Gty&index=34

YouTube IN Search Sign in

Directives in Angular

COMPONENTS	ATTRIBUTE DIRECTIVE	STRUCTURAL DIRECTIVE
Component is a directive with a template. E.g. - AppComponent	Attribute directive can be used as an attribute to change the behaviour or appearance of an element, component or other directives. ngStyle & ngClass	Structural directive controls the DOM element, using which we can add or remove elements from DOM. ngIf & ngFor

0:54 / 12:16

Angular Lifecycle Hook in Action | Lifecycle Hook |...
@ContentChild in Angular | Data Binding | Angular 12+
Custom Attribute Directive | Directives | Angular 12+
Renderer2 in Angular | Directives | Angular 12+
@HostListener in Angular | Directives | Angular 12+
@Input Binding in Angular |...

Custom Attribute Directive | Directives | Angular 12+ procademy

34 / 105

procademy

57.3K subscribers Join Subscribe 335 Share Download ...

Type here to search

Angular 12+ COMPLETE COURSE 2022

Custom Attribute Directive | Directives | Angular 12+ procademy

10:55 07-11-2024

Untitled - Paint

File Home View

Cut Copy Paste Select Crop Rotate Tools Brushes Shapes Colors Size

app CustomDirective setbackground.directive.ts

```
export class SetBackgroundDirective{}
```

here we created oneFolder with name CustomDirective and oneFile is created with ts file we know that ts file means it is a class file so take one class and to use this class outside of the file means then export it to make it directive use @Directive attribute

```
import { Directive, ElementRef } from '@angular/core';
@Directive({
  selector: '[setBackground]'
})
export class SetBackgroundDirective{
  constructor(element: ElementRef){
    element.nativeElement.style.backgroundColor = '#C8E6C9';
  }
}
```

app.component.html

```
<div class="container" setBackground>
  <p>This is a demo HTML content to understand custom attribute directive in Angular.</p>
</div>
```

app.module.ts

```
@NgModule({
  declarations: [
    AppComponent,
    SetBackgroundDirective
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

import that comp in APP.module folder

op:

This is a demo HTML content to understand custom attribute before op:

This is a demo HTML content to understand custom attribute op:

here the bg-color of the text has changed to green so here we created a simple CUSTOM directive which changes the bg-color of an element which we use it but here we are writing logic in constructor it is not a good practice write logic in ngOnInit() see below dig for more info

1 2 3 4 5 6

Untitled - Paint

File Home View

Cut Copy Paste Select Crop Rotate Tools Brushes Shapes Colors Size

app.component.html setbackground.directive.ts app.module.ts

```
app > CustomDirective > setbackground.directive.ts > SetBackgroundDirective > ngOnInit
1 import { Directive, ElementRef, OnInit } from '@angular/core';
2
3 @Directive({
4   selector: '[setBackground]'
5 })
6 export class SetBackgroundDirective implements OnInit{
7
8   constructor(private element: ElementRef){
9     //element.nativeElement.style.backgroundColor = '#C8E6C9';
10    this.element = element;
11  }
12
13  ngOnInit(){
14    this.element.nativeElement.style.backgroundColor = '#C8E6C9';
15  }
16 }
```

app.component.html

```
<app>
  <h1 setBackground>This is App Component</h1>
  <div class="container" setBackground>
    <p>This is a demo HTML content to understand custom attribute directive in</p>
  </div>
```

This is App Component

op:

This is a demo HTML content to understand custom attribute directive

1354 x 599px 11:26 07-11-2024

Services

The native element property contains the reference to the underlying DOM object which gives us direct access to the DOM, bypassing the angular. This is not advisable for following reasons:

- Angular keeps the Component & the view in Sync using Templates, data binding & change detection, etc. All of them are bypassed when we update the DOM Directly.
- DOM Manipulation works only in Browser. You will not be able to use the App in other platforms like in a web worker, in Server or in a Desktop, or in the mobile app, etc. where there is no browser.
- The DOM APIs do not sanitize the data. Hence it is possible to inject a script, thereby, opening our app an easy target for the XSS injection attack.

Renderer2 in Angular | Directives | Angular 12+

procademy 57.3K subscribers Join Subscribe 321 Share Download ...

Type here to search

Angular 13+ CHARTERS COURSE

Angular Lifecycle Hook in Action | Lifecycle Hook | procademy

@ContentChild in Angular | Data Binding | Angular 12+ procademy

Custom Attribute Directive | Directives | Angular 12+ procademy

Render2 in Angular | Directives | Angular 12+ procademy

@HostListener in Angular | Directives | Angular 12+ procademy

25°C Haze 11:29 07-11-2024

withOut services

```
<div class="container">
  <div style="text-align: center;">
    <h3>{{title}} Course</h3> js.html file
  </div>
  <div style="text-align: center; padding: 20px 0px;">
    <button (click)="OnEnroll()">Enroll</button>
  </div>
</div>
```

so when user click on enroll btn then alert will come

so here manan both angular and js comp's lo OnEnroll() event same logic 2 times rasam in future if we have 100 comps means it is very difficult to overcome that we will use services

angular.html

```
<div class="container">
  <div>
  </div>
  <div style="text-align: center;">
    <h3>{{title}} Course</h3> angular.html
  </div>
  <div style="text-align: center; padding: 20px 0px;">
    <button (click)="OnEnroll()">Enroll</button>
  </div>
</div>
```

angular.ts file

```
export class AngularComponent{
  title = "Angular"
  OnEnroll(){
    alert('Thank you for enrolling to ' + this.title)
  }
}
```

JavaScript Course

localhost:4200 says Thank you for enrolling to Angular course.

Angular Course

Enroll

26°C Haze 11:58 07-11-2024

create one service file. **service** is a class in angular without any decorators

```

src > app > Services > enroll.service.ts > EnrollmentService > OnEnrollClicked
1 export class EnrollmentService{           service file [1]
2   @OnEnrollClicked(title: string){        |
3     alert('Thank you for enrolling to '+title+' course.');?>
4   }
5 }

import { Component, OnInit } from '@angular/core';
import { EnrollmentService } from '../Services/enroll.service';

@Component({
  selector: 'app-javascript',
  templateUrl: './javascript.component.html',
  styleUrls: ['./javascript.component.css']
})
export class JavaScriptComponent{
  title = "JavaScript";
  to use services in comp then create
  the object for service file and use it
  OnEnroll(){
    const enrollmentService = new EnrollmentService();
    enrollmentService.OnEnrollClicked(this.title);
  }
}

```

create a object for service and use that objRef to access it

```

export class AngularComponent{
  title="Angular"
  angular.ts file
  OnEnroll(){
    const enrollmentService = new EnrollmentService();
    enrollmentService.OnEnrollClicked(this.title);
  }
}

```

here we are creating obj
that means tightly
coupled to make loosely
coupled use
dependencyInj see below

The screenshot shows a Microsoft Paint window with several code snippets and a diagram. The code snippets include `enroll.service.ts` (containing a service class), `javascript.ts` (containing a component that uses the service), and `angular.ts` (containing a component that injects the service). A red box highlights the import statement in `javascript.ts` and the creation of a service instance in its constructor. Another red box highlights the import statement in `angular.ts` and the injection of the service via the constructor. A third red box highlights the `objRef` usage in `angular.ts`. A fourth red box highlights the explanatory text: 'here we are creating obj that means tightly coupled to make loosely coupled use dependencyInj see below'. To the right of the code, there is a diagram showing two components: 'JavaScript Course' and 'Angular Course'. Each has an 'Enroll' button. A central box labeled 'JS' is connected to both courses, representing how the service is shared between them.

angular tasks for beginners - YouTube

YouTube IN

Services in Angular

Dependency Injection is a technique in which a class receives its dependencies from external sources rather than creating them itself.

Complete Angular 13 Course Step by...

procademy - 47 / 105

45 Services in Angular | Services & Dependency Injection | procademy

46 Creating a Service in Angular | Services & Dependency... procademy

47 Dependency injection in Angular | Services &... procademy

48 Hierarchical Injection in Angular | Services &... procademy

49 Creating a Data Service | Services & Dependency... procademy

50 Injecting Service into another

The screenshot shows a YouTube video player for a course titled 'Services in Angular'. The video is at 2:17 / 7:43. The video content shows a slide with an info icon and text about dependency injection. To the right, a sidebar displays a list of 105 video lessons, with the first few visible. The sidebar also includes a progress bar and navigation controls. The YouTube interface at the bottom shows the channel 'procademy' with 57.3K subscribers, and various sharing and download options.

```

app > javascript > javascript.components.ts > JavaScriptComponent > 
1 import { Component, OnInit } from '@angular/core';
2 import { EnrollService } from '../Services/enroll.service';
3
4 @Component({
5   selector: 'app-javascript',
6   templateUrl: './javascript.component.html',
7   styleUrls: ['./javascript.component.css']
8 })
9 export class JavascriptComponent{
10   title = "JavaScript";
11
12   constructor(private enrollService: EnrollService){
13   }
14
15   OnEnroll(){
16     enrollService.OnEnrollClicked(this.title);
17   }
18 }

so here by using constructor we created instance but angular don't know how to inject it. for that we will use Providers:[] so that angular will know how to inject

```

```

app > javascript > javascript.components.ts > JavaScriptComponent > 
1 import { Component, OnInit } from '@angular/core';
2 import { EnrollService } from '../Services/enroll.service';
3
4 @Component({
5   selector: 'app-javascript',
6   templateUrl: './javascript.component.html',
7   styleUrls: ['./javascript.component.css'],
8   providers: [EnrollService]
9 })
10 export class JavascriptComponent{
11   title = "JavaScript";
12
13   constructor(private enrollService: EnrollService){
14   }
15
16   OnEnroll(){
17     this.enrollService.OnEnrollClicked(this.title);
18   }
19 }

here we are not creating obj angular only creating obj and that objRef we are using to access fileds which are in service class this is called DI

```

ANGULAR 12+

COMPLETE COURSE 2022

Hierarchical Injection

Hierarchical Injection in Angular | Services & Dependency Injection | Angular 12+

procademy

procademy - 48 / 105

- Creating a Service in Angular | Services & Dependency... procademy
- Dependency injection in Angular | Services &... procademy
- Hierarchical Injection in Angular | Services &... procademy
- Creating a Data Service | Services & Dependency... procademy
- Injecting Service into another Service | Services &... procademy
- Component Interaction with...

procademy 57.3K subscribers Join Subscribe 325 Share Download ...

29°C Sunny 12:31 07-11-2024

Observable

We use **Observables** to perform asynchronous operations and handle asynchronous data.

The video player interface shows a yellow slide with the text above. Below it is a navigation bar with icons for back, forward, volume, and progress. The progress bar indicates 0:16 / 14:15. To the right of the video player is a sidebar titled "Complete Angular 13 Course Step by..." which lists several course modules from procademy.

What is Observable | Observables | Angular 12+

procademy 57.3K subscribers

Type here to search

Angular 13+ Complete Course

Error & Completion of Observable | Observables |...

procademy

29°C Sunny 12:32 07-11-2024

Untitled - Paint

File Home View

Cut Copy Paste Select Crop Rotate Tools Shapes Colors

```
1 import { Component, OnInit } from '@angular/core';
2 import { Observable } from 'rxjs';
3
4 @Component({
5   selector: 'app-root',
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.css']
8 })
9 export class AppComponent implements OnInit{
10   title = 'AngularObservables';
11   // create obj for observable
12   myObservable = new Observable((observer) =>{
13     console.log('Observable starts')
14     observer.next("1")
15     observer.next("2")
16     observer.next("3")
17     observer.next("4")
18     observer.next("5")
19   });
20   // to access observable we
21   // need to do subscribe
22   this.myObservable.subscribe((val) => {
23     console.log(val);
24   });
25 }
```

subscribe() method will take 3 optional parameters i) next, ii) error, iii) complete this 3 are optional and this 3 are callBack functions

op: all this data is notEmitted it is streamed

now to stream the data after some timeInterval then use

```
myObservable = new Observable((observer) =>{
  console.log('Observable starts')
  setTimeout(()=>(observer.next("1")), 1000)
  setTimeout(()=>(observer.next("2")), 2000)
  setTimeout(()=>(observer.next("3")), 3000)
  setTimeout(()=>(observer.next("4")), 4000)
  setTimeout(()=>(observer.next("5")), 5000)
  // observer.next("5")
})
```

based on time op will come

12:45 07-11-2024

