

# 1.What are RxJs in Angular ?

RxJS (Reactive Extensions for JavaScript) is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code.

## 2.Explain string interpolation and property binding in Angular.

string interpolation---- **one-way databinding** technique which is used to output the data from a TypeScript code to HTML template (view). It uses the template expression in **double curly braces** to display the data from the component to the view. String interpolation adds the value of a property from the component. **For example:**`{{ data }}`

Property Binding --- **one-way data binding** technique. In property binding, we bind a property of a DOM element to a field which is a defined property in our component TypeScript code. Actually Angular internally converts string interpolation into property binding.**For example:**`<img [src]="imgUrl" />`

### 3.How are observables different from promises?

Promises deal with one asynchronous event at a time. it means promises will be executed only once, even if we call `then()` multiple times.

Observables handle a sequence of asynchronous events over a period of time.

Observables are multicast, which means every time we subscribe to the observable, it will be executed again and again ( observables can be multicasted to multiple subscriptions)

## 4.Explain the concept of Dependency Injection?

Dependency Injection, or DI, is a design pattern and mechanism for creating and delivering some parts of an application to other parts of an application that require them. Angular supports this design pattern and you can use it in your applications to increase flexibility and modularity.

In Angular, dependencies are typically services, but they also can be values, such as strings or functions. An injector for an application (created automatically during bootstrap) instantiates dependencies when needed, using a configured provider of the service or value.

## 5. What are pipes in Angular explain with an example?

Pipes are simple functions to use in [template expressions](#) to accept an input value and return a transformed value. Pipes are useful because you can use them throughout your application, while only declaring each pipe once. ex: `{{ (true ? 'true' : 'false') | uppercase }}`

## 6.What exactly is a parameterized pipe?

In Angular, we can pass any number of parameters to the pipe using a colon (:) and when we do so, it is called Angular Parameterized Pipes. The syntax to use Parameterized Pipes in Angular Application is given below.

## 7.What are class decorators

Angular provides us with some class decorators. They allow us to tell Angular a particular class is a component or a module, e.g., And the decorator allows us to define this effect without putting any code inside the class.

An @Component and @NgModule decorator used with classes

## 8. What are Method decorators?

A Method Decorator decorates specific methods within your class with functionality. This is declared just before a method declaration. From TypeScript's documentation: The decorator is applied to the Property Descriptor for the method, and can be used to observe, modify, or replace a method definition



## 9. What are property decorators?

Property decorators are used to decorate the specific properties within the classes. Take a look at `@Input()`. Imagine that you have a property within the class that you want to have an input binding. Without decorators, you would have to define this property in your class for TypeScript to know about it, and then somewhere else tell Angular that you've got a property that you want to be an input.

# 10. What is the Component Decorator in Angular?

Component decorator allows you to mark a class as an Angular component and provide additional metadata that determines how the component should be processed, instantiated and used at runtime. Components are the most basic building block of an UI in an Angular application.

# 11. What are lifecycle hooks in Angular?

## Explain a few lifecycle hooks.

allow us to tap into the lifecycle of our components and trigger actions at specific points in the lifecycle.

- **ngOnChanges**: Called every time a data-bound input property changes. It's called a first time before the **ngOnInit** hook. The hook receives a SimpleChanges object that contains the previous and current values for the data-bound inputs properties. limit the amount of processing it does.
- **ngOnInit**: Called once upon initialization of the component.
- **ngDoCheck**: Use this hook instead of **ngOnChanges** for changes that Angular doesn't detect. It gets called at every change detection cycle, so keeping what it does to a minimum is important for performance.
- **ngAfterContentInit**: Called after [content is projected](#) in the component.
- **ngAfterContentChecked**: Called after the projected content is checked.
- **ngAfterViewInit**: Called after a component's view or child view is initialized.
- **ngAfterViewChecked**: Called after a component's view or child view is checked.
- **ngOnDestroy**: Called once when the component is destroyed and a good hook to use to cleanup and unsubscribe from observables.

## 12.What are router links?

In Angular, RouterLink is a directive for navigating to a different route declaratively. Router. navigate and Router. navigateByUrl are two methods available to the Router class to navigate imperatively in your component classes.

## 13.What exactly is the router state?

Angular RouterState is the state of the router as a tree of activated routes. It tells how the various components of an application are arranged on the screen to define what should be displayed on it. RouterState represents the state of the router as it keeps changing over time when users navigate from page to page.

# 14.What does Angular Material means?

Angular Material is a User Interface (UI) component library that developers can use in their Angular projects to speed up the development of elegant and consistent user interfaces. Angular Material offers you reusable and beautiful UI components like Cards, Inputs, Data Tables, Datepickers, and much more.

## 15. What is ngOnInit?

A callback method that is invoked immediately after the default change detector has checked the directive's data-bound properties for the first time, and before any of the view or content children have been checked. It is invoked only once when the directive is instantiated

## 16.What is transpiling in Angular ?

Transpiling is **converting one higher level language to another higher level language**. For example, Typescript is a high level language but after it's transpiled it's turned into JavaScript (another high level language).