

# Top Interview Questions and And Answers Of The Angular: By Pradip

- **Question No:- 1 :- Whats the use of Angular ?**
  - Angular is UI binding framework which binds the HTML UI and JS model.
  - This helps you to reduce your effort on writing those lengthy lines of code for binding.
  - adding to it , it also help you to build SPA by using concept of routing.
  - Its also has lot of other features like HTTP, DI, Input output because of which you do not need other framework
- **Question No:- 2 :- What are directives in Angular ?**
  - Directives are markers on HTML DOM file elements that tell angular to attach a behaviour to that element.
  - They are a way to extend, transform, and manipulate the DOM and its behavior within Angular applications.
- **Question No:- 3 :- Explain the different types of Angular directives ?**
  - Directives can be classified into two main types: structural directives and attribute directives.
  - Structural Directives
  - Structural directives are responsible for manipulating the structure of the DOM by adding, removing, or replacing elements. They are prefixed with an asterisk (\*) in the template syntax.
  - Angular provides built-in directives, such as ngIf, ngFor, and ngSwitch, which are used for conditional rendering, looping through arrays, and switching between multiple views, respectively.
  - Additionally, you can also create custom directives to encapsulate and reuse behavior across components.
  - Attribute Directives
- **Question No:- 4 :- Explain the importance of NPM and Node\_Modules folder ?**
  - Node package manager, which makes installation of javascript framework easily.
  - node\_modules is the folder where all the packages are installed
- **Question No:- 5 :- Explain the importance of Package.json file in Angular ?**
  - It has all the javascript references needed for project. So rather than installing on package at a time we can install all package in one go.
  - Whatever js framework you are using in a project they are all listed in package.json.
- **Question No:- 6 :- What is typescript and why do we need it ?**
  - As a name says it adds types to javascript. Typescript is a superset of JS.
  - It gives a nice object-oriented programming environment which transpiles / converts to JS.
  - So as it strongly typed we will have less errors and because we can do OOP with JS our productivity and quality also increases.
- **Question No:- 7 :- Explain importance of Angular CLI ?**
  - It stands for Command line interface
  - We can create initial angular project template. SO rather than starting from the scratch we have some boiler plate code
  - to install cli the command is - npm install @angular/cli
- **Question No:- 8 :- Explain the importance of Component and Modules ?**
  - Components is where you write your binding code. Module logically groups of components.
- **Question No:- 9 :- What is a decorator in Angular ?**

- Decorators are special types of declarations in TypeScript that are used to modify the structure or behavior of classes or class members. Angular comes with a set of built-in decorators that are used to configure and enhance various elements in an Angular application.
- Decorators defines what kind of class is it.
- @Component: Used to define a component and its metadata.
- @Directive: Used to define a directive and its metadata.
- @Injectable: Used to define a service and its metadata for dependency injection.
- @NgModule: Used to define a module and its metadata.
- The decorator function receives information about the decorated item and can modify its behavior, add metadata, or perform other tasks.
- The @Component decorator is commonly used to define Angular components. It takes a metadata object as an argument, which provides information about the component.
- Question No:- 10 :- What are Annotationa or MetaData ?**
  -
- Question No:- 11 :- What is a template ?**
  - Template is a HTML view of angular in which we can write directives
  - There are two way of defining template one is inline and other is seperate HTML file
- Question No:- 12 :- Explain the four types of Data bindings in Angular ?**
  - Data binding defines how the view and component communicate with each other. There are four types of the data binding in angular,
  - Interpolation binding {}- data flows from the component to the view and we can mix the same with html tags
  - property binding [] - data flows from the component to the view
  - event binding () - when you want to send event to the component
  - Two way binding [] - data flows from the component to the view and vice-versa
- Question No:- 13 :- Explain architecture of Angular ?**
  -
- Question No:- 14 :- What is SPA in Angular ?**
  - SPA stands for single page application
  - SPAs are applications where the main UI gets loaded once and then the needed UI is loaded on demand
  - In SPAs, the initial HTML, CSS, and JavaScript resources are loaded when the user first accesses the application.
  - Unlike traditional multi-page applications, SPAs do not require a full page reload with each user interaction. Instead, subsequent interactions or changes in content are handled by dynamically updating the existing page through asynchronous requests.
- Question No:- 15 :- How to implement SPA in Angular ?**
  -
- Question No:- 16 :- How to implement routing in Angular ?**
  - Routing is a simple collection which has two things URL and when this URL called which component to load.
  - So routing help you to define the navigation for your angular application
  - So if you want to move from one screen to other screen and you want to respect SPA that means not loading and refreshing the whole UI routing is needed.
- Question No:- 17 :- Explain Lazy Loading ?**
  -
- Lazy loading is a technique used in web development to optimize the loading time of a web page by deferring the loading of certain resources until they are actually needed.
- In the context of routing in a web application, lazy loading refers to loading specific modules

or components only when the user navigates to a particular route, rather than loading all modules and components at the initial page load.

- Lazy loading routing is commonly used in Single Page Applications (SPAs) where the entire application is loaded initially, but certain parts of the application are loaded on-demand as the user interacts with the application.
- **Question No:- 18 :- How to implement Lazy Loading in Angular ?**
  - Lazy loading involves importing the component only when it's needed. In a lazy-loaded setup, the About Us component would be loaded only when the user navigates to the '/about-us' route.
  - When lazy loading a component, you typically use dynamic imports. Dynamic imports return a Promise, and the component is loaded when the Promise is resolved.
  - Note: If you see `import()` or something similar in your code, it's a good indication that lazy loading is being used.

```
{  
  
  path: 'about-us',  
  
  loadChildren: () => import('./features/about-us/about-us.module').then(m  
  
=> m>AboutUsModule)  
  
}
```

- path: 'about-us':

This specifies the route path. In this case, it's 'about-us'. This means that when the user navigates to the 'about-us' route, the specified module (lazy-loaded) will be loaded and associated with this route.

- loadChildren: () => import('./features/about-us/about-us.module').then(m => m>AboutUsModule):

This is the key part that enables lazy loading. Instead of directly importing the module at the time the application loads, it uses the loadChildren property to specify a function that will be called when the module is needed.

- `import('./features/about-us/about-us.module')` is a dynamic import statement that returns a Promise. The specified module is not loaded immediately; it will be loaded asynchronously when the Promise is resolved.
  - The `.then(m => m>AboutUsModule)` part ensures that the module is loaded and the specific module class (`AboutUsModule`) is retrieved. This is essential because Angular expects a dynamically loaded module to have a certain structure, and this structure is defined by the module class.
  - In summary, when a user navigates to the 'about-us' route, the associated module (`AboutUsModule`) will be loaded asynchronously, reducing the initial bundle size and improving the application's loading performance. Lazy loading is especially useful in large applications where loading all modules at once would lead to slower initial load times.
- **Question No:- 19 :- Define Services ?**
    - Services are used for organizing and sharing code across components.
    - They are singleton objects that can be injected into components, providing a way to encapsulate and share functionality.
  - **Question No:- 20 :- What is Dependency Injection ?**
    - DI is an application design pattern where rather than creating object instances from within the component, Angular injects it via the constructor.
    - Dependency injection in Angular allows you to declare the dependencies of a class (like services) in its constructor, and Angular's dependency injection system will provide instances of those dependencies when the class is instantiated.
    - Angular's dependency injection system is used to inject services into the components or other services that need them. When a component or another service requests a dependency (like a service) in its constructor, Angular provides the instance of that dependency.
  - **Question No:- 21 :- How to implement Dependency Injection ?**
    -
  - **Question No:- 23 :- What's the benefit of Dependency Injection ?**
    - This makes the code more modular, testable, and maintainable, as dependencies can be easily swapped or mocked during testing. The `@Injectable` decorator is an essential part of this mechanism, allowing Angular to understand how to create and manage instances of services.
    - DI helps to decouple class dependencies, so that when you add new dependencies do not have to change everywhere..
  - **Question No:- 24 :- Differentiate between `ng serve` and `ng build` ?**
    - `ng serve` builds angular applications in memory or in ram, while `ng build` builds the applications on the hard disk.
    - So when you want to go for the production '`ng build`' command is used.
  - **Question No:- 25 :- Explain the `--prod` parameter in `ng build` ?**
    - `Ng build --prod` flag compresses your JS file, removes comments, creates GUIDs of your JS files and makes your application ready for production.
  - **Question No:- 26 :- Explain `ViewChild` and `ViewChildren`?**
    - If we want to access DOM elements that are present inside the HTML Template of the same or child component. In that case, we can use `View Child` and `View Children` Decorators.
    - Also, if we want to access child component properties and methods, that is possible by using `View Child` and `View Children` Decorators.
  - **Question No:- 27 :- Why do we need Template reference variables?**

- Template reference variables give you a way to declare a variable that references a value from an element in your template.
- A template reference variable can refer to the following: a DOM element within a template (including custom elements) an Angular component or directive.