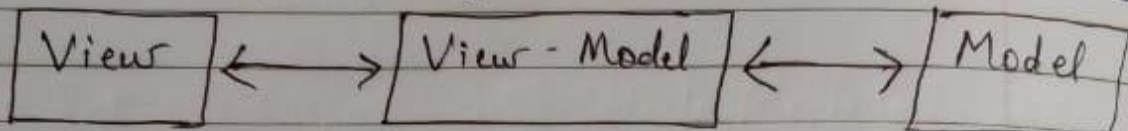


# Interview Questions

Q 1. What is Angular & tell its uses?

Ans. Angular is a Javascript binding framework which binds the HTML, UI & javascript model.

This helps you to reduce your effort on writing lengthy lines of code for binding.



HTML

Angular

Object

View-model is the binding code which is taken care by Angular.

It also helps you to build SPA by using the concept of routing. It also has a lot other features like HTTP, Dependency Injection, Input, Output because of which you do not need other frameworks.

Q

Angular JS

	Angular JS	Angular
Language	Javascript	Typescript
Architecture	MVC Controller	MVVM Component
CLI	No	Yes
Lazy loading	No	Yes
Mobile Complaint	No	Yes
SEO	No	Yes
Version	1	2 and above

3, 4, 5

## Q2 What are directives in Angular?

Ans Directives are defined as classes that can add, remove and modify the existing behaviours of the elements in the template. The purpose of directive is to maneuver the DOM.

We can say, Directives are Angular syntaxes which we write inside HTML. ex. `[(ngModel)]`  
~~value~~  
`[hidden]`

## Q2 Different types of Angular Directives

Ans There are 3 types of directives in Angular.

1. Structural
2. Attribute
3. Component

### 1. Structural Directive

change the DOM structure by adding or removing elements. Like

\* `ngFor = "let item of school"`

### 2. Attribute Directive

They change the appearance and behaviour of HTML elements. They do not add or remove elements, do not change structure of DOM. Like `ngModel`, `ngSwitch`, `ngClass`, `ngStyle`

~~`<div [hidden] = "hide()">Hello</div>`~~



### 3. Component Directive

are directive that include a template that is rendered in the place of component. They are directives with ~~component~~ templates.

In component directives, three main parameters are there:-

1. Selector
2. template Url
3. style Urls

Q What is npm and Node Modules folder?

A NPM stands for Node Package Manager. It makes installation of javascript framework easy. For ex.  
 npm install jquery

### Node Modules

is the folder where all the packages are installed locally.

Q Importance of Package. JSON file

Ans. Package.json file is the file in which all the references of javascript framework used in the application are listed down. Rather than installing a package at a time, we can install all packages in one go using

npm install



Q What is TypeScript & what is its benefit?  
 Ans. Typescript is a superset of Javascript. It extends JavaScript by adding types to the language.

Benefits :-

1. It speeds up the development by catching errors & providing fixes before you even run your code.
2. It makes Javascript strongly typed language.
3. Also it gives nice object oriented programming environment which transpiles/converts to Javascript.

Q Importance of Component & Modules

Ans Component

Components are the main building blocks of an Angular Application. It is a reusable unit formed by a template and a class that controls a section of the UI.

The class contains attributes & methods that describe component's behaviour. It is defined using @Component decorator.

Modules

A module refers to a place where we can group the components, services, directives and pipes.

It is defined by @NgModule



Q What is Angular CLI?

Ans CLI stands for Command Line Interface. It is a command line tool that is designed to aid developers in creating, building, testing and deploying Angular applications efficiently. It helps developers to create projects easily and quickly. Rather than starting from scratch, we have some boiler plate code.

Q What are Annotations, Metadata, Decorator in Angular?

Ans To create a component in Angular, you define a Typescript class & annotate it with the @Component decorator.

@Component is the annotation.

It provides metadata, about <sup>to Angular</sup> ~~the~~ ~~component~~ a class, ~~its~~ such as its selector, template <sup>etc</sup> and how ~~it should be treated~~ a class or its members <sup>should be treated</sup>.

Some other annotations are

@ Ng Module

@ Pipe

@ Directive

@ Injectable

@ View Child

Annotation and Decorators are the same things.



## Q2 Different types of Data Binding in Angular

Ans Data Binding

- Data binding is a way to bind the data (variables, objects, arrays) defined in your Typescript code (ie model or business logic) to your HTML (View).

It defines how the view and component communicate with each other.

It establishes a connection b/w app UI (view) and the data it displays.

When the data changes its value, the elements ~~are~~ that are bound to data reflect changes automatically.

There are four types of Data binding in Angular

1. Interpolation
2. Property Binding
3. Event Binding
4. Two way Binding

### 1. Interpolation -

is a way to transfer data from a typescript code to an HTML template

ex.

{{ value }}

### 2. Property Binding -

helps you set values for properties of HTML elements or directives.

ex.

[hidden] = "hide()"

Property Binding

It is an example of one-way binding.

[innerText] =

<div ~~width~~ = "100\*50" >

Data flows from component class to property of a HTML element.

### 3. Event Binding

In Event Binding information or data flows from the view to the component when an event is triggered.

ex. mouse click

(click) = "updateName ()"

### 4. Two Way Binding

In two way binding, there is two way movement of data, i.e. from view to component & vice-versa. We use a directive ngModel to achieve this binding.

ex.

<input [(ngModel)] = "text">

Also we need to import a Forms module from '@angular/forms' into app. module. Is

import {FormsModule} from '@angular/forms'

other equivalent syntax:

<input [ngModel] = "text" (ngModelChange) = "text" = \$event >



Q What is SPA?

Ans SPA stands for Single Page Application.

In SPA's main UI gets loaded once and then the needed UI is loaded on demand. We don't load all UI again and again but we just load the required part only.

For ex.

a typical website could have different sections like Navbar, Footer, left sidebar, content section.

So first three sections are loaded only once and the content section UI will only changes & loaded from server as the user navigates. This is achieved with the help of routing concept.

Q What is lazy Loading?

Ans Lazy loading means on demand loading. In this technology, Angular allows you to load components when they are needed or a specific route is activated. It reduces the application loading time by splitting the application into many bundles.

# There must be at least 2 <sup>or more</sup> modules in the application.

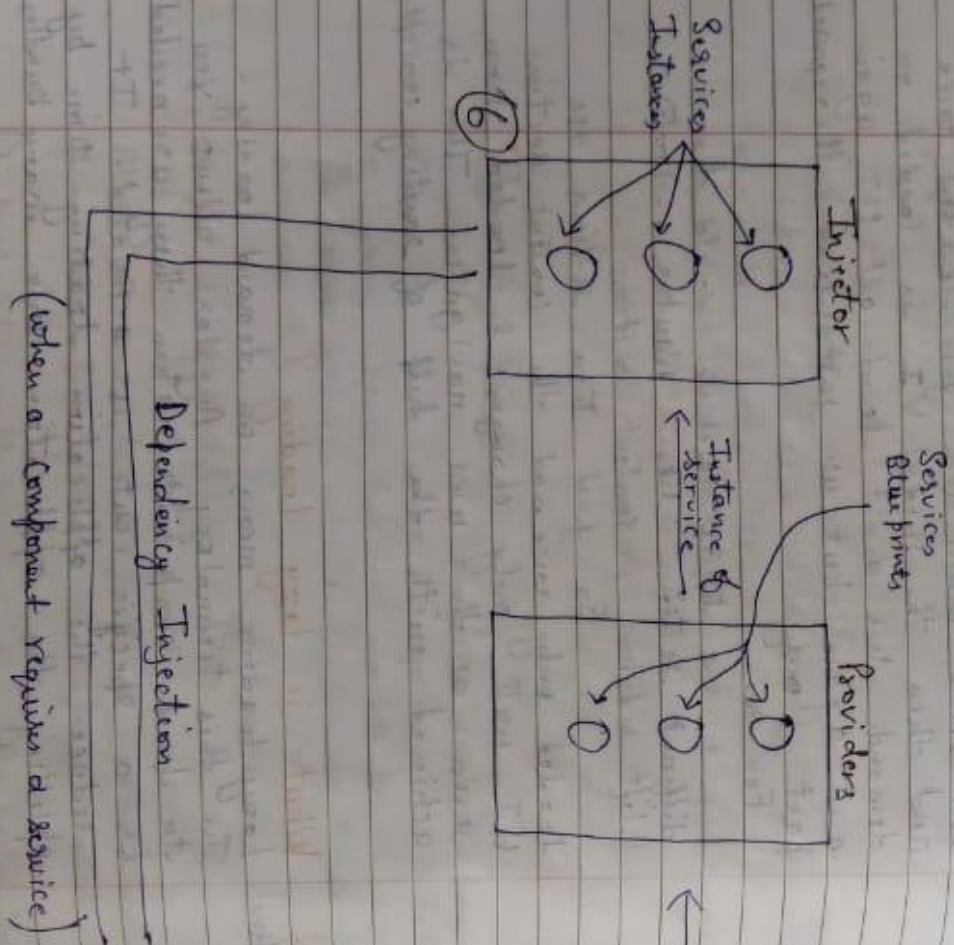
These modules are loaded dynamically when a user navigate to a route associated with that module. We use `loadChildren` property instead of `component` while configuring routes.



output

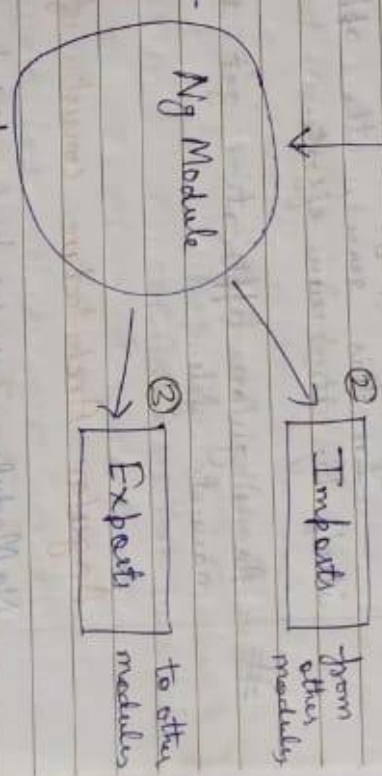
Ng module A will be generated a both added in module A's child: 'a', lead children: module A's app module

# ANGULAR ARCHITECTURE



Starting Point or Entry point or Bootstrapping of ng App

Main.ts



Declarations





Q What is Bootstrapping in software?

Ans Bootstrapping is the process in which involves a chain of stages, in which at each stage, a relatively small and simple program loads and then executes the larger, more complicated program.

In this sense, the application improves itself by its own efforts.

# An Angular application, get started by main.ts file.

Angular Architecture consists of

Ng Module is a core part which handles compilation context

- Imports
- Exports
- Declarations
- Providers
- Component
- Templates
- Directives
- Services
- Pipes
- Dependency Injection



## Q What is Routing?

Ans. Routing allows the users to create SPA's with multiple views and allows navigation between them.

Angular provides a separate module, the router Module, for setting up navigations.

```
export const HomeRoutes = [
  { path: "", component: HomeComponent },
  { path: "login", component: LoginComponent, canActivate: [AuthGuard] }
]
```

## Q What is Dependency Injection?

Ans. Dependency Injection is a design-pattern & mechanism for creating & delivering some parts of an application to other parts.

It allows classes, components, modules to be interdependent while maintaining consistency.

— When a component requests a service, the request goes to the Injector, which keeps <sup>instance</sup> ~~record~~ of services. If Injector has, instance of that service, it directly provides. Otherwise it asks to provider and then new instance is created, stored in the Injector registry and provided.



## Advantages of DI

1. Boiler plate code is reduced, as initializing of dependencies is done by the Injector component.
2. Extending the application becomes more manageable.
3. It enables loose coupling while maintaining consistency.

## If not using DI, then

1. Code will not be flexible. Anytime dependencies change, the dependent classes need to be changed as well.
2. Code is not suitable for testing. Anytime you instantiate a service, a new instance will get created & you will get fresh values even if they are changed earlier.  
As a result we will not have control over code.

Q.

### Ng Serve

1. It builds in RAM
2. It uses JIT compilation

### Ng build

- It builds on the hard-disk & creates dist folder.  
It uses AOT compilation.

Q. What does ng build --prod?

Ans. It produces most compressed code of your application by removing comments, creates GUID's of your JS files.

So, whenever we want to deploy our code to production, we must use this command.



Q.

Observable

Promise

Ans.

1. Observables can emit multiple values over time.

Promise can emit only single value.

2. Observables are lazy. It will execute only after subscribing it.

Promises are not lazy. They execute immediately.

3. Observables can be synchronous or asynchronous.

Promises are asynchronous.

4. Observable can be unsubscribed.

Promises can't be unsubscribed.

5. Observables have different RxJS functions like filter, map, reduce etc.

Promises don't have any function.

Q. What is View Model and Template?

Ans. Template - In Angular, template refers to DOM elements, directives. It is the HTML part.

Model - Model is the component instance to which the template belongs.

View - template + Model

The template and the model together form view of a component. It is basically UI part which is visible on screen.



Q What is JIT?

Ans JIT stands for Just-In-Time. It is a type of compilation in which compilation is done at the runtime ~~just~~ before the execution of the program. i.e. code gets compiled when it is needed, not ~~at the~~ whole application's code gets compiled at the build time like in AOT.

Q What is AOT?

Ans. AOT stands for Ahead-Of-Time. The AOT compiler converts the code during build phase long before the browser download and runs the code.

### Advantages of AOT

1. Faster Rendering -

As the code is already pre-compiled, so it gets rendered directly on the browser.

2. Smaller Angular framework Download Size -

As the app is already compiled, so there is no need to download Angular compiler. It dramatically reduces the application <sup>payload</sup>.

3. Earlier Detection of template Errors

binding errors are reported during build time. Template



#### 4. Fewer Asynchronous Requests

The compiler inlines the external HTML templates and CSS style sheets within application at build time.

#### 5. Better Security

AOT compiles HTML templates and components into JS files long before they are served to clients. There are fewer opportunities for injection attacks.

Note. In Angular .json file

~~build~~ Configuration Projects  
demo  
architect  
build  
Configuration  
production  
aot: true

ng build --aot  
ng serve --aot

# upto Angular 8 - JIT  
Angular 9 and above - AOT

## JIT

1. It downloads the Angular compiler and compiles the code exactly before displaying in the browser.
2. Loading of app is slower.
3. More suitable for development ~~process~~ mode.
4. Bundle size is bigger.
5. Template binding error can be ~~not~~ caught at display time.
6. Command  
ng serve  
ng build

## AOT

- AOT doesn't download compiler as whole code gets compiled at the build time.
- Loading of app is faster.
- More suitable for production mode.
- Bundle size is smaller.
- Template binding error can be caught at build time.
- Command  
ng serve --aot  
ng build --aot