Angular Interview Questions

1. What’s new in Angular 5?

Certain tools are optimized in the new version of [Angular](https://www.greycampus.com/angularjs-training-instructor-led), let us see what the tools are:

* Angular 5 supports Typescript version 2.4
* Angular 5 supports RxJS 5.5 which has new features like Pipeable Operators
* A build tool to make the js bundles (files) lighter
* Ahead of Time (AOT) is updated to be on by default
* Events like ActivationStart and ActivationEnd are introduced in Router

2. Name the building blocks of Angular.

The Angular application is made using the following:

* Modules
* Component
* Template
* Directives
* Data Binding
* Services
* Dependency Injection
* Routing

3. What is Transpiling in Angular?

Transpiling is the process of converting the typescript into JavaScript. Though typescript is used to write code in the Angular applications, the code is internally transpiled into JavaScript.

4. Which of the Angular life cycle component execution happens when a data-bound input value update?

ngOnChanges is the life cycle hook that gets executed whenever a change happens to the data that was bound to an input.

5. Differentiate between Components and Directives in Angular 5.

Components break up the application into smaller parts; whereas, Directives add behavior to an existing DOM element.

6. What is the use of @Input and @Output?

When it comes to the communication of Angular Components, which are in Parent-Child Relationship; we use @Input in Child Component when we are passing data from Parent to Child Component and @Output is used in Child Component to receive an event from Child to Parent Component.

7. What is ng-content Directive?

The HTML elements like p (paragraph) or h1 (heading) have some content between the tags. For example, <p>this is a paragraph</p> and <h1>this is a heading</h1>. Now, similar to this, what if we want to have some custom text or content between the angular tags like <app-tax>some tax-related content</app-tax> This will not work the way it worked for HTML elements.  Now, in such cases, the <ng-content> tag directive is used.

8. What does a router.navigate do?

When we want to route to a component we use router.navigate.

Syntax: this.router.navigate([‘/component\_name’]);

9. What is ViewEncapsulation?

ViewEncapsulation decides whether the styles defined in a component can affect the entire application or not. There are three ways to do this in Angular:

Emulated: styles from another HTML spread to the component.

Native: styles from another HTML do not spread to the component.

None: styles defined in a component are visible to all components.

10. What are Services in Angular and what command is used to create a service?

Services help us in not repeating the code. With the creation of services, we can use the same code from different components. Here is the command to create a service in angular, ng g service User (a UserService is created when this command is used).

11. What is Dependency Injection in Angular 4?

When a component is dependent on another component the dependency is injected/provided during runtime.

12. What is Routing in Angular 5?

Routing helps a user in navigating to different pages using links.

13. How to handle Events in Angular 5?

Any activity (button click, mouse click, mouse hover, mouse move, etc) of a user on a frontend/web screen is termed as an event. Such events are passed from the view (.HTML) page to a typescript component (.ts).

14. What is a RouterOutlet?

RouterOutlet is a substitution for templates rendering the components. In other words, it represents or renders the components on a template at a particular location.

15. Explain the usage of {{}}?

The set of brackets {{}} when used with an HTML tag, represent data from a component. For example, on a HTML page which has <h1>{{variableName}}</h1>, here the ‘variableName’ is actually typescript (component) data representing its value on the template; i.e., HTML. This entire concept is called String Interpolation.

16. In how many ways the Data Binding can be done?

Data Binding happens between the HTML (template) and typescript (component). Data binding can be done in 3 ways:

1. Property Binding (ii) Event Binding (iii) Two-Way Data Binding.

17. What is the sequence of Angular Lifecycle Hooks?

OnChange()  -  OnInit()  -  DoCheck()  -  AfterContentInit()  -  AfterContentChecked()  -  AfterViewInit()  -  AfterViewChecked()  -  OnDestroy().

18. What is the purpose of using package.json in the angular project?

With the existence of package.json, it will be easy to manage the dependencies of the project. If we are using typescript in the angular project then we can mention the typescript package and version of typescript in package.json.

19. How is SPA (Single Page Application) technology different from the traditional web technology?

In traditional web technology, the client requests for a web page (HTML/JSP/asp) and the server sends the resource (or HTML page), and the client again requests for another page and the server responds with another resource. The problem here is a lot of time is consumed in the requesting/responding or due to a lot of reloading. Whereas, in the SPA technology, we maintain only one page (index.HTML) even though the URL keeps on changing.

20. What is Component in Angular Terminology?

A web page in Angular has many components involved in it. A Component is basically a block in which the data can be displayed on HTML using some logic usually written in typescript.

21. What are ngModel and how do we represent it?

ngModel is a directive which can be applied on a text field. This a two-way data binding. ngModel is represented by [()]

22. What does a Subscribe method do in Angular 4?

It is a method which is subscribed to an observable. Whenever the subscribe method is called, an independent execution of the observable happens.

23. Differentiate between Observables and Promises.

Observables are lazy, which means nothing happens until a subscription is made. Whereas Promises are eager; which means as soon as a promise is created, the execution takes place. Observable is a stream in which passing of zero or more events is possible and the callback is called for each event. Whereas, promise handles a single event.

24. What is an AsyncPipe in Angular?

When an observable or promise returns something, we use a temporary property to hold the content. Later, we bind the same content to the template. With the usage of AsyncPipe, the promise or observable can be directly used in a template and a temporary property is not required.

25. Explain Authentication and Authorization.

Authentication: The user login credentials are passed to an authenticate API (on the server). On the server side validation of the credentials happens and a JSON Web Token (JWT) is returned. JWT is a JSON object that has some information or attributes about the current user.  Once the JWT is given to the client, the client or the user will be identified with that JWT.

Authorization: After logging in successfully, the authenticated or genuine user does not have access to everything. The user is not authorized to access someone else’s data, he/she is authorized to access some data.

26. What is AOT Compilation?

An Angular application consists mainly of components and their HTML templates. Because the components and templates provided by Angular cannot be understood by the browser directly, Angular applications require a compilation process before they can run in a browser.

The Angular Ahead-of-Time (AOT) compiler converts your Angular HTML and TypeScript code into efficient JavaScript code during the build phase before the browser downloads and runs that code.

27. What is Redux?

It is a library which helps us maintain the state of the application. Redux is not required in applications that are simple with the simple data flow, it is used in Single Page Applications that have complex data flow.

28. What are the Pipes?

This feature is used to change the output on the template; something like changing the string into uppercase and displaying it on the template. It can also change Date format accordingly.

29. Differentiate between ng-Class and ng-Style.

In ng-Class, loading of CSS class is possible; whereas, in ng-Style we can set the CSS style.

30. Why Typescript with Angular?

Typescript is a superset of JavaScript. Earlier, JavaScript was the only client-side language supported by all browsers. But, the problem with JavaScript is, it is not a pure Object-Oriented Programming Language. The code written in JS without following patterns like Prototype Pattern becomes messy and finally leading to difficulties in maintainability and reusability. Instead of learning concepts (like patterns) to maintain code, programmers prefer to maintain the code in an OOP approach and is made available with a programming language like Typescript was thus developed by Microsoft in a way that it can work as JavaScript and also offer what JavaScript cannot i.e.

* pure OOPS as Typescript offers concepts like Generics, Interfaces and Types (a Static Typed Language) which makes it is easier to catch incorrect data types passing to variables.
* TS provides flexibility to programmers experienced in java, .net as it offers encapsulation through classes and interfaces.
* JS version ES5 offers features like Constructor Function, Dynamic Types, Prototypes. The next version of JavaScript i.e. ES6 introduced a new feature like Class keyword but not supported by many browsers.
* TS offers Arrow Functions (=>) which is an ES6 feature not supported by many browsers directly but when used in TS, gets compiled into JS ES5 and runs in any browser.
* TS is not the only alternative to JS, we have CoffeeScript, Dart (Google).
* Finally, it is like, TS makes life easier when compared to JS.

# 31. What is the meaning of <base href="/"> in angular

The **<base** href="/"**>** tells the Angular router what is the static part of the URL. The router then only modifies the remaining part of the URL.

Ex: <base href="/subdomain">

In browser it looks like: <http://localhost:4500/subdomain/comp1>

32. What is the use of @NgModule

Inside of the @NgModule operator, we define all the properties of the module. For that, we provide a simple JavaScript object as the parameter. Let's take a closer look, at what each property of that object actually does.

33. What is the use of Bootstrap

Defines the root-component of the Application. Only use this in the AppModule.

34. What is the use of Exports

We define the components, directives or pipes we want to export here. That means, that our module is providing these to other modules when they get imported. Otherwise, these components stay module internal and cannot be accessed from the outside.

35. What is the use of Declarations

Inside of the declarations array, we define all the components, directives and pipes, that are declared and used inside this module. If a component (or directive or pipe) is not added to the declarations array and you use it in your module/application, angular will throw an error at runtime. Also, a component (or ... you got it) can only be declared in one module. If you want to use your component in multiple modules, you need to bundle that component into a separate module and import that in the module.

36. What is the use of Imports

Speaking of importing... Your module can import as many sub-modules as you like. Don't have defined any custom modules yet? No problem, we will get to that.

But even if you don't have any modules, you still need to import some angular modules. As I mentioned earlier, Angular is built with modularity in mind. While many features are contained in angular's core, some features are bundled into their own module. For example, if you want to use the HttpClient, you will need to import the HttpClientModule.

37. What is the use of Providers

We define any @Injectables, required by the module, here. Any sub-components or modules can then get the same instance of that @Injectable via dependency injection. In the case of the AppModule, these @Injectables are application-scoped.

38. What is the use of ng-template

ng-template is used to not render the data when page loads. It renders only when the ngIf condition will be true or the template loads when the by using the reference of the template.

Ex: <ng-template #template> (Show when template is true) </ng-template>

39. What is the use of ng-container.

ng-container is used to place two structural directives (\*ngIf, \*ngFor) at one place. It renders automatically when the page loads.

40. Explain Angular Authentication and Authorization.

The user login credentials are passed to an authenticate API, which is present on the server. Post server-side validation of the credentials, a JWT (JSON Web Token) is returned. The JWT has information or attributes regarding the current user. The user is then identified with the given JWT. This is called authentication.

Post logging-in successfully, different users have a different level of access. While some may access everything, access for others might be restricted to only some resources. The level of access is authorization.

41. What are the different types of filters in Angular?

Below are the various filters supported by Angular:

* currency: Format a number to a currency format.
* date: Format a date to a specified format.
* filter: Select a subset of items from an array.
* json: Format an object to a JSON string.
* limit: To Limits an array/string, into a specified number of elements/characters.
* lowercase: Format a string to lower case.
* number: Format a number to a string.
* orderBy: Orders an array by an expression.
* uppercase: Format a string to upper case.

42. Differentiate between one-way binding and two-way data binding.

One-way data binding is achieved using the double curly braces {{}} or square braces [] or \*.

Whereas, in Two-way data binding, the View or the UI part is updated implicitly as soon as the data model changes. It is a synchronization process, unlike One-way data binding.

43. What are the lifecycle hooks for components and directives?

An Angular component has a discrete life-cycle which contains different phases as it transits through birth till death.

constructor: It is invoked when a component or directive is created by calling new on the class.

ngOnChanges: It is invoked whenever there is a change or update in any of the input properties of the component.

ngOnInit: It is invoked every time a given component is initialized. This hook is only once called in its lifetime after the first ngOnChanges.

ngDoCheck: It is invoked whenever the change detector of the given component is called. This allows you to implement your own change detection algorithm for the provided component.

ngOnDestroy: It is invoked right before the component is destroyed by Angular. You can use this hook in order to unsubscribe observables and detach event handlers for avoiding any kind of memory leaks.

44. What are events in Angular?

Events in Angular are specific directives that help in customizing the behavior of various DOM events. Few of the events supported by Angular are listed below:

* click
* copy
* cut
* dblclick
* keydown
* keypress
* keyup
* mousedown
* mouseenter
* mouseleave
* mousemove
* mouseover
* mouseup
* blur

45. What is bootstrapping in Angular?

Bootstrapping in Angular is nothing but initializing, or starting the Angular app. Angular supports automatic and manual bootstrapping.

46. What is Angular Framework?

Angular is a TypeScript-based open-source front-end platform that makes it easy to build applications with in web/mobile/desktop.

47. What is a data binding?

Data binding is a core concept in Angular and allows to define communication between a component and the DOM. There are four forms of data binding (divided as 3 categories) which differ in the way the data is flowing.

* Interpolation: {{value}}:
* Property binding: [property]=”value”:
* Event binding: (event)=” function”
* Two-way data binding: [(ngModel)]=”value”

48. What is HttpClient?

Most of the Front-end applications communicate with backend services over HTTP protocol. Angular provides a simplified client HTTP API known as HttpClient which is available from @angular/common/http package. You can import in your root module as below,

Ex: import {HttpClientModule} from '@angular/common/http';

49. How do you perform Error handling?

If the request fails on the server or failed to reach the server due to network issues then HttpClient will return an error object instead of a successful response. In this case, you need to handle in the component by passing error object as a second callback to subscribe () method.

50. What is the purpose of Wildcard route?

If the URL doesn't match any predefined routes then it causes the router to throw an error and crash the app. In this case, you can use wildcard route.

Ex: {path: '\*\*', component: PageNotFoundComponent}

51. What are different types of compilation in Angular?

Angular offers two ways to compile your application,

* Just-in-Time (JIT)
* Ahead-of-Time (AOT)

52. What is JIT?

Just-in-Time (JIT) is a type of compilation that compiles your app in the browser at runtime. JIT compilation is the default when you run the ng build (build only) or ng serve (build and serve locally) CLI commands. Views take longer to render because of the in-browser compilation step.

53. What are all the metadata properties of NgModule? And what are they used for?

@NgModule accepts a metadata object that tells Angular how to compile and launch the application. The properties are:

imports – Modules that the application needs or depends on to run like, the BrowserModule that every application needs to run in a browser.

declarations –We must declare every component class in an NgModule class. If we use a component without declaring it, we'll see a clear error message in the browser console.

bootstrap – The application will be launched by creating the components listed in this array.

54. Write some code for a basic class in TypeScript with a constructor and a method.

class Test {

constructor () {

this.sampleMethod();

}

sampleMethod() {

for (let i = 0; i < 10; i++) {

console.log(i)

}

}

}

let test = new Test ();

55. What‘s the basic syntax of a Decorator in Angular?

@() with optional parameters.

56. What function is called when an object is created in TypeScript? What is it‘s basic syntax in TypeScript code?

The constructor function is called. It’s syntax is:  Constructor(){}

57. What modules should you import in Angular to use [(ngModel)] and reactive forms?

FormsModule and Reactiveforms Module.

58. How similar is AngularJS to Angular 2?

Both are front-end frameworks maintained by Google, but Angular 2 is not a simple update of AngularJS, it is a new framework written from scratch. Updating an app from AngularJS to Angular 2 would require a complete rewrite of the code.

59. What were some features introduced in the different versions of Angular (2, 4, 5 and 6)?

Angular 2:

* Complete rewrite of the Angular framework
* Component-based rather than controllers/view/$scope. This allows more code to be reused, easier communication between components and easier testing
* Much faster
* Support for mobile devices
* More language choices such as TypeScript

Angular 4:

* An update to Angul
* ar 2, not a complete rewrite. Updating from Angular 2 to 4 just requires updating the core libraries
* Improvements to Ahead-of-time (AOT) generated code
* Support for new versions of TypeScript
* Animation packages are removed from the core package
* Else block

Angular 5:

* Focused on making Angular smaller and faster to use
* Http is depreciated and HttpClient API client is now recommended for all apps
* Supports TypeScript 2.3
* Introduction of a build optimizer
* Angular Universal State Transfer API
* Improvements to the Angular Compiler
* Router Lifecycle Events
* Better cross-browser standardization

Angular 6:

* Better service worker support
* Better URL serialization
* Ivy rendering engine
* ng update and ng add
* <template> element completely removed
* Angular Elements/Custom Elements
* Form validation changes
* Schematics

60. What is CLI?

CLI is the acronym of Command Line Interface, which can be used to create the Angular JS application. Using CLI, you can also create a unit and end-to-end tests for the Angular application.

61. Why are decorators used in Angular 2?

In Angular 2, decorators are used as an identifier of class or type of the object that is created by the TypeScript.

62. How to get params value in Angular.

import { ActivatedRoute } from '@angular/router';

constructor(public actRoute:ActivatedRoute){}

this.actRoute.params.subscribe(res => {

console.log(res)

})

63. How will you convert a string into a percentage?

To convert a string into a percentage format, a percent filter is used.

Ex: {{courseDuration | percent}}

64. Explain the concept of lazy loading in Angular 2.

Lazy loading is a module which is used to decrease the start-up time. When lazy is used, then our system application does not need to load everything at once. It only needs to load what the user expects to see when the application first loads. The modules which are lazily loaded will only be loaded when the user navigates to their routes.

65. When to use Ngoninit and constructor in Angular 2?

Constructors are used for initializing class members and also for dependency injection. Ngonlnit is used for the initialization work. Both of these methods are called when the component is created.

66. What are Event emitters?

An Event emitter is a class defined in core module that can be used by components and directives to emit custom events.

67. What is a Cookie?

A cookie is a small piece of data sent from a website and stored on the user's machine by the user's web browsers while the user is browsing.

68. What Is Parameterizing Pipe?

A pipe can accept any number of optional parameters to achieve output. The parameter value can be any valid template expressions. To add optional parameters, follow the pipe name with a colon (:). Its looks like- currency: 'INR'

Ex: <h2>The birthday is - {{birthday | date:"MM/dd/yy" }} </h2>

<! -- Output - The birthday is - 10/03/1984 -->

69. What is Chaining Pipe?

The chaining Pipe is used to perform the multiple operations within the single expression. This chaining operation will be chained using the pipe (I).

In the following example, to display the birthday in the upper case- will need to use the inbuilt date-pipe and upper-case-pipe.

Ex: {{birthday | date | uppercase}}

70. Why You Use Browsermodule, Commonmodule, Formsmodule, Routermodule, And Httpclientmodule?

BrowserModule – The browser module is imported from @angular/platform-browser and it is used when you want to run your application in a browser.

CommonModule – The common module is imported from @angular/common and it is used when you want to use directives - NgIf, NgFor and so on.

FormsModule – The forms module is imported from @angular/forms and it is used when you build template driven forms.

RouterModule – The router module is imported from @angular/router and is used for routing RouterLink, forRoot, and forChild.

HttpClientModule –The HttpClientModule is imported from @angular/common/http and it used to initiate HTTP request and responses in angular apps. The HttpClient is more modern and easier to use the alternative of HTTP.

71. What are types of directives in Angular?

In angular there are 4 types of directives. They are

* Component directives
* Structural directives
* Attribute directives
* Custom directives

Component directives: It is mainly used to specify the HTML templates. It is the most commonly-used directive in an Angular project. It is decorated with the @component decorator. This directive is a class.

Structural directives: The structural directive is used to add or remove the HTML Element in the Dom Layout. Its built-in types are \*NgIf, \*NgFor, \*NgSwitch. Structural directives are easy to recognize by using an asterisk (\*)

Attribute directives: Attribute directive is used to change/modify the behavior of the HTML element in the Dom Layout. Its built-in type is [NgStyle], [NgClass].

Custom directives: We can create custom attribute directives and custom structural directives using a @Directive decorator.

Ex: import { Directive, ElementRef } from '@angular/core';

@Directive({

selector: '[myRed]'

})

export class MyRedDirective {

constructor(elRef: ElementRef) {

elRef.nativeElement.style.color = 'red';

}

}

72. What is the use of @component decorator.

It provides metadata that determines how the component should be processed, instantiated and used at runtime.