INTERVIEW QUESTIONS

1. What are pipes?

A pipe takes in data as input and transforms it to a desired output. You can chain pipes together in potentially useful combinations. You can write your own custom pipes. Angular comes with a stock of pipes such as DatePipe, UpperCasePipe, LowerCasePipe, CurrencyPipe, and PercentPipe.

2. What is the minimum definition of a component?

The absolute minimal configuration for a @Component in Angular is a template. Both template properties are set to optional because you have to define either template or templateUrl.

3. What's the difference between an Angular component and module?

Components control views (html). They also communicate with other components and services to bring functionality to your app.

Modules consist of one or more components. They do not control any html. Your modules declare which components can be used by components belonging to other modules, which classes will be injected by the dependency injector and which component gets bootstrapped. Modules allow you to manage your components to bring modularity to your app.

4. How can I select an element in a component template?

You can get a handle to the DOM element via ElementRef by injecting it into your component's constructor

5. What is an observer?

Observer is an interface for a consumer of push-based notifications delivered by an Observable. It has below structure,

```
interface Observer<T> {
    closed?: boolean;
    next: (value: T) => void;
    error: (err: any) => void;
    complete: () => void;
```

```
  }
```

A handler that implements the Observer interface for receiving observable notifications will be passed as a parameter for observable as below,

```
myObservable.subscribe(myObserver);
```

6. What is an observable?

An Observable is a unique Object similar to a Promise that can help manage async code. Observables are not part of the JavaScript language so we need to rely on a popular Observable library called RxJS. The observables are created using new keyword. Let see the simple example of observable,

7. What is Redux and how does it relate to an Angular app?

Redux is a way to manage application state and improve maintainability of asynchronicity in your application by providing a single source of truth for the application state, and a unidirectional flow of data change in the application. ngrx/store is one implementation of Redux principles.

8. What are the Core Dependencies of Angular 7?

RxJS 6.3, TypeScript 3.1

9. Why Incremental DOM Has Low Memory Footprint?

Virtual DOM creates a whole tree from scratch every time you rerender.

Incremental DOM, on the other hand, doesn’t need any memory to rerender the view if it doesn’t change the DOM. We only have to allocate the memory when the DOM nodes are added or removed. And the size of the allocation is proportional to the size of the DOM change.

10. What are the ways to control AOT compilation?

1. By providing template compiler options in the tsconfig.json file
2. By configuring Angular metadata with decorators

11. What is activated route?

ActivatedRoute contains the information about a route associated with a component loaded in an outlet. It can also be used to traverse the router state tree. The ActivatedRoute will be injected as a router service to access the information.

12. What is router outlet?

The RouterOutlet is a directive from the router library and it acts as a placeholder that marks the spot in the template where the router should display the components for that outlet. Router outlet is used as a component,

13. What are the utility functions provided by RxJS?

The RxJS library also provides below utility functions for creating and working with observables.

1. Converting existing code for async operations into observables
2. Iterating through the values in a stream
3. Mapping values to different types
4. Filtering streams
5. Composing multiple streams

14. What is multicasting?

Multi-casting is the practice of broadcasting to a list of multiple subscribers in a single execution. Let's demonstrate the multi-casting feature

15. What is subscribing?

An Observable instance begins publishing values only when someone subscribes to it. So you need to subscribe by calling the subscribe() method of the instance, passing an observer object to receiving the notifications.

16. What is the difference between *ngIf vs [hidden]?

*ngIf effectively removes its content from the DOM while [hidden] modifies the display property and only instructs the browser to not show the content but the DOM still contains it.

17. What is the difference between "@Component" and "@Directive" in Angular?

Directives add behaviour to an existing DOM element or an existing component instance.

- A component, rather than adding/modifying behaviour, actually creates its own view (hierarchy of DOM elements) with attached behaviour.

Write a component when you want to create a reusable set of DOM elements of UI with custom behaviour. Write a directive when you want to write reusable behaviour to supplement existing DOM elements.

18. How would you protect a component being activated through the router?

The Angular router ships with a feature called guards. These provide us with ways to control the flow of our application. We can stop a user from visiting certain routes, stop a user from leaving routes, and more. The overall process for protecting Angular routes:

- Create a guard service: ng g guard auth
- Create canActivate() or canActivateChild() methods

19. What is the difference between Structural and Attribute directives in Angular?

Structural directives are used to alter the DOM layout by removing and adding DOM elements. It is far better in changing the structure of the view. Examples of Structural directives are NgFor and Nglf.

- Attribute Directives These are being used as characteristics of elements. For example, a directive such as built-in NgStyle in the template Syntax guide is an attribute directive.

20. What is the purpose of base href tag?

The routing application should add element to the index.html as the first child in the tag inorder to indicate how to compose navigation URLs. If app folder is the application root then you can set the href value as below

21. What is a bootstrapping module?

Every application has at least one Angular module, the root module that you bootstrap to launch the application is called as bootstrapping module. It is commonly known as AppModule.

22. What is interpolation?

Interpolation is a special syntax that Angular converts into property binding. It's a convenient alternative to property binding. It is represented by double curly braces({{}}). The text between the braces is often the name of a component property. Angular replaces that name with the string value of the corresponding component property.

23. Explain the difference between `Promise` and `Observable` in Angular?

Promise emits a single value while Observable emits multiple values. So, while handling a HTTP request, Promise can manage a single response for the same request, but what if there are multiple responses to the same request, then we have to use Observable.

24. Explain the difference between "Constructor" and "ngOnInit"?

The Constructor is a default method of the class that is executed when the class is instantiated and ensures proper initialisation of fields in the class and its subclasses. Angular, or better Dependency Injector (DI), analyses the constructor parameters and when it creates a new instance by calling new MyClass() it tries to find providers that match the types of the constructor parameters, resolves them and passes them to the constructor

ngOnInit is a life cycle hook called by Angular to indicate that Angular is done creating the component.

We have to import OnInit like this in order to use it (actually implementing OnInit is not mandatory but considered good practice):

25. What is difference between "declarations", "providers" and "import" in NgModule?

imports makes the exported declarations of other modules available in the current module

declarations are to make directives (including components and pipes) from the current module available to other directives in the current module. Selectors of directives, components or pipes are only matched against the HTML if they are declared or imported.

providers are to make services and values known to DI (dependency injection). They are added to the root scope and they are injected to other services or directives that have them as dependency.

26. Features of Angular 7 are:

CLI Prompts, Application Performance, Virtual Scrolling, Drag and Drop, Angular Compiler, Angular Do-Bootstrap, Better Error Handling

27. What is a template in Angular7?

In Angular 7, templates are written with the help of HTML that contains Angular JS specification elements and attributes. The main role of Angular 7 is to combine the templates which ultimately control the dynamic view that is visible to the users in the browsers.

28. What is AOT?

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. ... The browser loads executable code so it can render the application immediately, without waiting to compile the app first.

29. What is Angular Universal?

Angular Universal is the process of server-side rendering (SSR) your application to HTML on the Server (ie: Node. js). Typical Angular applications are Single-Page Applications (aka SPA's) where the rendering occurs on the Browser. This process can also be referred to as client-side rendering (CSR).

30. What is a parameterized pipe?

A pipe can accept any number of optional parameters to fine-tune its output. To add parameters to a pipe, follow the pipe name with a colon (:) and then the parameter value (such as currency:'EUR'). If the pipe accepts multiple parameters, separate the values with colons (such as slice:1:5)

31. What is Zone in Angular?

In simple language Zone.js is a api or set of programs which is used by angular 2 to update the application view when any change occurred. A Zone is an execution context that persists across asynchronous task. for example:Events, XMLHttpRequests and Timers(setTimeout(), setInterval()) etc.

32. What does a just-in-time (JIT) compiler do (in general)?

A Just-In-Time (JIT) compiler is a feature of the run-time interpreter, that instead of interpreting bytecode every time a method is invoked, will compile the bytecode into the machine code instructions of the running machine, and then invoke this object code instead.

33. Why would you use lazy loading modules in Angular app?

Lazy Loading is the technique of loading the module or data on demand. It helps us to better the application performance and reduce the initial bundle size of our files. The initial page loads faster and we can also split the application into the logic chunks which can be loaded on demand.

34. What are the advantages with AOT?

1. Faster download: - The Angular 2 app is already compiled so it is faster. 2. Faster Rendering: - If the app is not AOT compiled and the compilation process happens in the browser once the application is fully loaded.

35. What is the difference between pure and impure pipe?

A pure pipe is only called when Angular detects a change in the value or the parameters passed to a pipe. An impure pipe is called for every change detection cycle no matter whether the value or parameter(s) changes.

36. What is the difference between BehaviorSubject vs Observable?

Observable is a Generic, and BehaviorSubject is technically a sub-type of Observable because BehaviorSubject is an observable with specific qualities.

37. Why did the Google team go with incremental DOM instead of virtual DOM?

They have one goal in mind: applications have to perform well on mobile devices. This mainly meant optimizing two things: the bundle size and the memory footprint. ... The rendering engine has to have low memory footprint.

38. Explain the Architecture overview of Angular.

- Component
- Templates
- Metadata
- Data Binding
- Directives
- Services
- Dependency Injection

39. How would you update Angular 6 to Angular 7?

ng update @angular/cli @angular/core

40. What is the UrlSegment Interface in Angular 7?

The UrlSegment is a part of a URL between the two slashes and it contains a path and matrix parameters associated with the segment.

41. What is Do Bootstrap (ng Do Bootstrap) In Angular 7?

The ng Do Bootstrap is a new life-cycle hook added in Angular 7 and Do Bootstrap is an interface.

42. What is IVY Renderer? Is it supported by Angular 7?

Angular will be releasing a new kind of rendering pipeline and view engine. The purpose of angular view engine is to translate the templates and components that we have written into the regular HTML and JavaScript so it is easy for the browser to read it comfortably. Ivy is believed to be splendid for the Angular Renderer.

Yes, it is supported by Angular 7.

43. What is TestBed in Angular?

Angular Test Bed (ATB) is the primary API for writing unit tests for Angular Application and libraries. It allows us to test behaviors and change detections that depend on the Angular Framework

44. What Is Node.js?

Node.js is a powerful framework developed on Chrome's V8 JavaScript engine that compiles the JavaScript directly into the native machine code. It is a lightweight framework used for creating server-side web applications and extends JavaScript API to offer usual server-side functionalities. It is generally used for large-scale application development, especially for video streaming sites, single-page application, and other web applications.

45. List down the major benefits of using Node.js?

Fast, Asynchronous, Scalable, Open Source, No Buffering

46. What is an error-first callback in Node.js?

Error-first callbacks in Node.js are used to pass errors and data. The very first parameter you need to pass to these functions has to be an error object while the other parameters represent the associated data. Thus you can pass the error object for checking if anything is wrong and handle it. In case there is no issue, you can just go ahead and with the subsequent arguments.

47. Explain the purpose of module.exports?

A module in Node.js is used to encapsulate all the related codes into a single unit of code which can be interpreted by shifting all related functions into a single file. For example, suppose you have a file called greet.js that contains the two functions as shown below:

48. Explain the purpose of ExpressJS package?

Express.js is a framework built on top of Node.js that facilitates the management of the flow of data between server and routes in the server-side applications. It is a lightweight and flexible framework that provides a wide range of features required for the web as well as mobile application development. Express.js is developed on the middleware module of Node.js called connect. The connect module further makes use of http module to communicate with Node.js. Thus, if you are working with any of the connect based middleware modules, then you can easily integrate with Express.js.

49. What are Streams? List types of streams available in Node Js ?

- Readable − For reading operation.
- Writable − For writing operation.
- Duplex − Used for both read and write operation.
- Transform − A type of duplex stream where the output is computed based on the input.

50. Life Cycle hooks of Angular ?

1. ngOnChanges()
2. ngOnInit()
3. ngDoCheck()
4. ngAfterContentInit()
5. ngAfterContentChecked()
6. ngAfterViewInit()
7. ngAfterViewChecked()
8. ngonDestroy

51. What is Javascript Closure? Purpose?

A **closure** is the combination of a function bundled together (enclosed) with references to its surrounding state (the **lexical environment**). In other words, a closure gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created, at function creation time.

```
function init() {
  var name = 'Mozilla'; // name is a local variable created by init
  function displayName() { // displayName() is the inner function, a closure
    alert(name); // use variable declared in the parent function
  }
  displayName();
}
init();
```

52. Difference between Subject & Behaviour Subject?

BehaviourSubject will return the initial value or the current value on Subscription

Subject does not return the current value on Subscription. It triggers only on .next(value) call and return/output the value

	Each next subscribers receive...
`Subject`	...only upcoming values
`BehaviorSubject`	...one previous value and upcoming values
`ReplaySubject`	...all previous values and upcoming values
`AsyncSubject`	...latest value when stream will close

53. CSS Display Property?

Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element.

Also, with **display: inline-block**, the top and bottom margins/paddings are respected, but with display: inline they are not.

Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

54. Dependencies vs devDependencies?

Dependencies should contain libs and framewors your app is built on, such as Vue, React, Angular, Express, JQuery and etc. You will agree with me, if I say, that your project wont work without these packages

devDependencies should contain packages which are used during development or which are used to build your bundle, for example, mocha, jscs, grunt-contrib-watch, gulp-jade and etc. These packages are neseccery only while you are developing your project, also ESlint is used to check everything during building your bundle.

55. What is XSS?

Cross-site Scripting (XSS) is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application.

56. How can you secure your HTTP cookies against XSS attacks?

XSS occurs when the attacker injects executable JavaScript code into the HTML response.
To mitigate these attacks, you have to set flags on the set-cookie HTTP header:

HttpOnly - this attribute is used to help prevent attacks such as cross-site scripting since it does not allow the cookie to be accessed via JavaScript.
Secure - this attribute tells the browser to only send the cookie if the request is being sent over HTTPS.
So it would look something like this: Set-Cookie: sid=<cookie-value>; HttpOnly. If you are using Express, with express-cookie session, it is working by default

57. In JS we follow error-first callback, then why in promise, first func is resolve? why not reject?

Some callbacks can consume more parameters than just two, hence, trying to figure out if err is second, third etc would be confusing, therefore, having err passed as a first one makes sens.

58. What are the two types of API functions in Node.js ?

The two types of API functions in Node.js are

a) Asynchronous, non-blocking functions
b) Synchronous, blocking functions

59. Can you access DOM in node?

No, you cannot access DOM in node.

60. What are the two arguments that async.queue takes?

The two arguments that async.queue takes

a) Task function

b) Concurrency value

61. How Node.js overcomes the problem of blocking of I/O operations?

Node.js solves this problem by putting the event based model at its core, using an event loop instead of threads.

62. What is an event loop in Node.js ?

To process and handle external events and to convert them into callback invocations an event loop is used. So, at I/O calls, node.js can switch from one request to another .

63. Event loop stages ?

timers: this phase executes callbacks scheduled by setTimeout() and setInterval().

pending callbacks: executes I/O callbacks deferred to the next loop iteration.

idle, prepare: only used internally.

poll: retrieve new I/O events; execute I/O related callbacks (almost all with the exception of close callbacks, the ones scheduled by timers, and setImmediate()); node will block here when appropriate.

check: setImmediate() callbacks are invoked here.

close callbacks: some close callbacks, e.g. socket.on('close', ...).

64. What is ‘Callback’ in node.js?

Callback function is used in node.js to deal with multiple requests made to the server. Like if you have a large file which is going to take a long time for a server to read and if you don’t want a server to get engage in reading that large file while dealing with other requests, call back function is used. Call back function allows the server to deal with pending request first and call a function when it is finished.

65. forRoot() vs forChild()

forRoot creates a module that contains all the directives, the given routes, and the router service itself.

forChild creates a module that contains all the directives and the given routes, but does not include the router service.

66. Child process in node js

https://www.freecodecamp.org/news/node-js-child-processes-everything-you-need-to-know-e69498fe970a/