**Angular Interview Questions:**

# What is angular?

* Angular is a Java script framework to build client-side applications.
* It is Great for single-page applications.
* Refresh automatically without reloading the webpage.
* Angular promotes a modular approach (application build having clear structure).
* Making use of components we have a lot of reusable code.
* Angular has a lot of inbuilt capabilities such as routing, validation, etc which makes the development easier and quicker.
* Angular is a product of Google and uses the typescript language for Microsoft.

# Difference B/w Angular JS and angular

|  |  |  |
| --- | --- | --- |
| Feature | AngularJS | Angular |
| Architecture | Supports MVC design model | Uses components and directives |
| Language | Recommended Language: JavaScript | Recommended Language: TypeScript |
| Expression Syntax | Specific ng directive is required for the image/property and an event | Uses () to bind an event and [] for property binding |
| Mobile Support | Doesn’t provide any mobile support | Provides mobile support |
| Routing | $routeprovider.when() is used for routing configs | @RouteConfig{(…)} is used for routing config |
| Dependency Injection | Doesn’t supports the concept of Dependency Injection | Supports hierarchical Dependency Injection with a unidirectional tree-based change detection |
| Structure | Less manageable | Simplified structure and makes the development and maintenance of large applications easier |
| Speed | With two-way data binding development effort and time are reduced | Faster than AngularJS with upgraded features |
| Support | No support or new updates are provided anymore | Active support and frequent new updates are made |

# Advantages of using Angular

* It supports two-way data-binding
* It follows MVC pattern architecture
* It supports static template and Angular template
* You can add a custom directive
* It also supports REST full services
* Validations are supported
* Client and server communication is facilitated
* Support for dependency injection
* Has strong features like Event Handlers, Animation, etc.

# Features in angular

* Angular cli
* Animation support
* Cross-platform app development
* Code splitting
* Templates
* Testing

# Angular History?

AngularJS

* AngularJS is also called angular 1
* It was introduced in 2010.
* Angular 1 is the first version of the angular framework.
* It was based on Java script language

Angular 2 --> 2016 sep-2

* It is completely rewritten the angular 1.
* It was introduced in the year of 2016 September.
* It was based on typescript
* Angular 2 is mobile-oriented
* The major drawback is not capable to support the mobile app

Angular3

* Angular 3 was skipped due to the misalignment of the router package version.

Angular 4

* Angular 4 was introduced in March 2017.
* Angular 4 is an Enhancement of angular 2.
  + inbuilt animation packages
  + ng template
  + form validation attributes
  + title case

Angular 5

* Angular 5 was Introduced in November 2017.
* Faster compared to the previous version of angular and smaller in size.
* Various features in angular 5
  + HTTP client API
  + multiple exports and alias
  + pipes for numbers, data, and currency
  + lambda support

Angular 6

* Angular 6 was introduced in May 2018.
* Features came with the release of angular 6
  + Updation of angular CLI ( command-line interface )
  + Updation of angular CDK (component development kit)

Angular 7

* Angular 7 was introduced in October 2018.
* Enhancement of Angular 6
* Feature of Angular 7 is:
  + Cli promotes
    - Helps the user makes a decision
    - It asks user want to add routing (y/N)
    - About style, user want to use like CSS/sass/less
  + Application performance
  + Virtual scrolling
  + Drag and drop 🡪 automatic rendering features
  + Bundle budget
    - Bundle size is more than 2MB
    - Waring msg provided for above 5 MB

Angular 8

* Angular 8 was introduced in May 2018.
* Dynamic imports for route configurations.
* Customize the angular CLI commands like **ng Build, ng test, ng run.**
* Ng deploy is added in the angular CLI 8.3.0

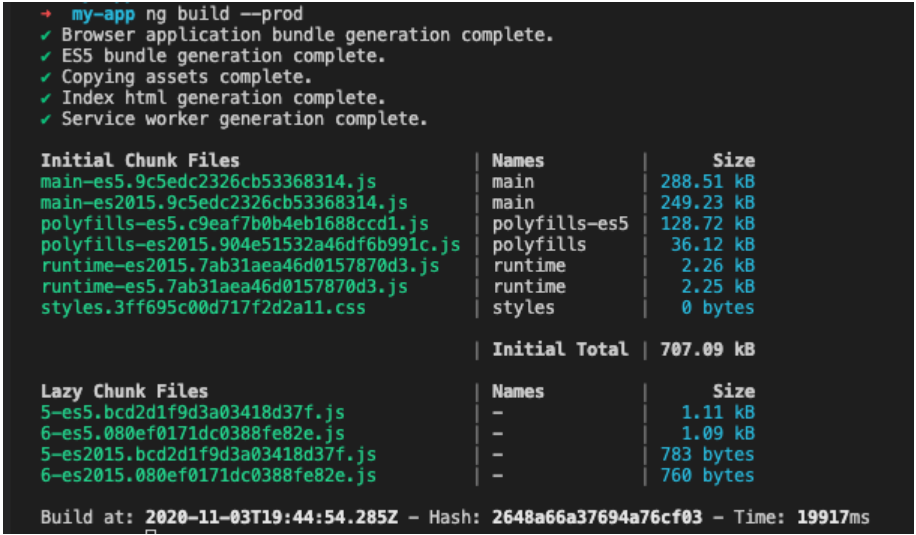
Angular 9

* Angular 9 was introduced in February 2020.
* Angular 9 came up with an **ivy compiler.**
  + From version 9 all applications are moved to ivy compiler and runtime by default.
  + It provided the following advantages.
    - Smaller Bundle Size
    - Faster Testing
    - Better Debugging
    - Improved CSS class and style binding
    - Improved Type Checking
    - Improved build errors
    - Improved build times, enabling AOT on by default
    - Improved Internationalization.
  + New Options for **provided-In** property in **@Injectable** Decorator, In addition to the previous root and module options, you have two additional options.
  + **Platform**: Specifying **provided-In: 'platform'** makes the service available in a special singleton platform injector that is shared by all applications on the page.
  + **Any** Provides a unique instance in every module (including lazy modules) that injects the token.

Angular 10

* Angular 10 was introduced in June 2020.
* Angular 10 is the smallest release compared to other releases
* TSLint has updated v2.0
* TSLib has updated v6
* New default browser configuration.
* Optional stricter setting
* Warnings about common.JS imports

Angular 11

* Angular 6 was introduced in November 2020.
* Automatic inline of fonts.
* Component test harnesses – provide a robust and legible API service.
* Improved reporting and logging.
  + We’ve made changes to the builder phase reporting to make it even more helpful during development.
  + 
* Updated hot module replacement

# Semantic versioning

Three major components (10.0.1)

Major (10)

Minor (0)

Patch (1)

# Summary of angular

* Angular apps are in modular in nature.
* Angular apps are collection of individual modules
* Angular app contain one or more modules.
* Module contain one or more components
* Components contain HTML + class to control the logic or view
* Service contain the business logic
* Module can import or exports the code as required and finally render the view in browser.

# Why were client-side frameworks like Angular introduced?

* Web developers used VenilaJS and jQuery to develop dynamic websites.
* For that applications that use complex logic, developers had to put in extra effort to maintain separation of concerns for the app.
* JQuery did not provide facilities for data handling across views.
* For tackling the above problems, client-side frameworks like Angular came into the picture,
* which made life easier for the developers by handling separation of concerns and dividing code into smaller bits of information(In the case of Angular, called Components)
* clientside frameworks allow to develop advance web applications like single page application
* we can also develop spas using venilajs but development process become slower

# How does an Angular application work?

-> Every Angular app consists of a file named angular.json.

-> This file will contain all the configurations of the app.

-> While building the app, the builder looks at this file to find the entry point of the application.

-> Inside the build section entry point of application is Main.ts

-> The main.ts file creates a browser environment to run the application, along with this it also calls a function bootstrap Module, which bootstraps the application

# Angular Lifecycle hooks.

constructor

-> **ngOnchanges** :

* This method is called whenever one or more input properties of component changes.

-> **ngOnInit** :

* This hook gets called once, after the ngonchanges hook.
* it initializes the component and sets the input properties of component

-> **ngDoCheck**

* It is called after **ngonchanges** and **ngonit**
* It is used to detect and act on changes that cannot be detect by angular

-> **ngAftercontentInit**

* it gets called after ngDocheck.
* This hook responds after the content get projected inside the component

-> **ngAftercontentcheck** -

* this method is called after ngaftercontentinit and every subsequent ngdocheck.
* it responds after the projected content is checked

-> **ngAfterViewInit** -

* it responds after component views or a child component views is initialized

-> **ngAfterviewChecked**

* it response after the component or child views is checked

-> **ngonDestroy**

* it gets called, just before angular destroy the component.
* This method is used to cleanup the code and detach the event handlers.

# Compilation types

**JIT(Just-in-Time) compilation** - In JIT compilation, the application compiles inside the browser during runtime. By default, angular builds and serves the application using JIT compiler:

* ng build
* ng serve

**AOT(Ahead-of-Time) compilation**- Whereas in the AOT compilation, the application compiles during the build time. For using AOT compiler following changes should be made:

Angular AOT (Ahead of Time) is a compiler that converts your angular HTML and typescript code into the JavaScript code.

* ng build --aot
* ng serve –aot

advantages of aot complilation

fast rendering, minimizing errors, better security

# Could we make an angular application to render on server side

Yes we can, with angular Universal, a technology provided by angular capable of rendering the application server side.

Benefits of angular Universal are:

* Better user experience
  + Allows users to see the view of application instantly.
* better SEO
  + universal ensures that content is available on every search engine leading for better seo
* Loads Faster
  + Render pages are available to the browser sooner, so the server side applications loads faster

# Explain Components, Modules and Services in Angular

Modules:

* A module is a place where we can group components, directives, services, and pipes.
* Angular apps are modular in nature
* Angular apps are collection of individual modules
* Every module is defined with @NgModule decorater.
* Every angular module contain one or more modules ie root module or app module

Components:

* Components are basic building blocks of which controls a part of UI for any application.

Services:

* The main objective of a service is to share data, functions with different components of an Angular application.

# Explain string interpolation and property binding in Angular.

* String interpolation and property binging are the part of databinding in angular
* Data-binding is a feature in angular, which provides a way to communicate between the component(Model) and its view(HTML template).
* Data-binding can be done in two ways, one-way binding and two-way binding.
* **String interpolation** and **property binding** allow only **one way data binding**.
* String interpolation uses the double curly braces {{ }} to display data from the component.
* Using property binding, we can bind the DOM properties of an HTML element to a component's property.
* Property binding uses the square brackets [ ] syntax.
* **Event Binding** – Enables the application to respond to user input in the target environment
* **Property Binding** – Enables interpolation of values computed from application data into the HTML
* **Two-way Binding** – Changes made in the application state gets automatically reflected in the view and vice-versa. The ngModel directive is used for achieving this type of data binding.

# How does one share data between components in angular?

* **Parent to child using @Input decorator**

@Component({

selector: 'app-parent',

template: `

<app-child [data]=data></app-child>

` ,

styleUrls: ['./parent.component.css']

})

export class ParentComponent{

data:string = "Message from parent";

constructor() { }

}

* In the above parent component, we are passing “data” property to the following child component

import { Component, Input} from '@angular/core';

@Component({

selector: 'app-child',

template:`

<p>{{data}}</p>

`,

styleUrls: ['./child.component.css']

})

export class ChildComponent {

@Input() data:string

constructor() { }

}

**Child to parent using @ViewChild decorator**

**Child to parent using @Output and EventEmitter**

Child Component:

import {Component, Output, EventEmitter} from '@angular/core';

@Component({

selector: 'app-child',

template:`

<button (click)="emitData()">Click to emit data</button>

`,

styleUrls: ['./child.component.css']

})

export class ChildComponent {

data:string = "Message from child to parent";

@Output() dataEvent = new EventEmitter<string>();

constructor() { }

emitData(){

this.dataEvent.emit(this.data);

}

}

As you can see in the child component, we have used **@Output** property to bind an **EventEmitter**. This event emitter emits data when the button in the template is clicked.  
  
In the parent component’s template we can capture the emitted data like this:

<app-child (dataEvent)="receiveData($event)"></app-child>

Then inside the receiveData function we can handle the emitted data:

receiveData($event){

this.dataFromChild = $event;

}

# Filters in Angular js

* currency – Formats a number to the currency format
* date – Formats a data to some specific format
* filter – Selects a subset of items from an array
* json – Formats an object to a JSON string
* limitTo – Limits an array or string into a specified number of characters or elements
* lowercase – Formats a string to lowercase
* number – Formats a number to a string
* orderBy – Orders an array by an expression

# What is scope in AngularJS

* Scopes are objects that refer to the model. They act as glue between the controllers and view.
* Two types of scopes
  + Local scope
  + Global scope

# Promise and observable

* Promise
  + A promise is java script object that links producing code and consuming code
    - Producing code – code that can take some time.
    - Consuming code – code that must wait for the result
  + Promise executes immediately
  + Promise is always synchronous
  + Holds single value
  + It is not lazy and eager
  + Cannot canceled by any method
* observable
  + observable handle notification
  + observable can be both synchronous or asynchronous
  + holds multiple values
  + it is lazy
  + can be cancelled using unsubscribe() method

# What is authentication

The authentication is a service that is used to login and logout of Angular application. The credentials of users pass to API on the server. Then post server-side validation these credentials, JSON Web Token is returned, which as detail about the current user.

# Directives

Component directives

* These directives have a view, a stylesheet and a selector property.
* Instead of @Directive decorator we use @Component decorator to declare these directives

Structural directives

* Ng if
  + \*ngIf is used to check a boolean value and if it’s truthy, the div element will be displayed.
* Ng for
  + \*ngFor is used to iterate over a list and display each item of the list.
* Ng Switch

Attribute directives

* Ng Class
* Ng Style
* Modify the attributes of Dom elements.
* These directives are used to change the look and behaviour of a DOM element

# Input and output decorators

* Parent to child using @Input decorator
* Child to parent using @View Child decorator
  + Child to parent using @Output and Event Emitter

# Dependency injection

* Dependency injection is an application design pattern which is implemented by Angular.
* It also one of the core concepts of Angular.
* Dependencies in angular are nothing but services which have a functionality.

# Async and await

* When an async function is called, it returns a promise.
* When async function returns a value. Promise will be resolved with returned value.
* When async function throws an exception or some value. promise will be rejected with thrown value.
* Async and await are the promises. And makes promises easier to write.

**Async 🡪** makes a function a return a value.

**Await 🡪** makes a function waits a promises

# Callbacks

A callback is a function passed as an argument to another function.

This technique allows a function to call another function

A callback function can run after another function has finished

# What is java script?

* Java script is used to create a client side dynamic web pages.
* Java script is a light weight and cross platform.
* Java script is object based scripting language
* Java script is not compiled language but it is a translated language.
* Java script translator can translate the java script code to web pages.
* Java script is used to create interactive websites.
* Mainly used for :
  + Client side validations
  + Displaying date and time
  + Display date and time ,clock
  + Display dropdowns and menus
  + Display popup windows and alerts
* Java script is faster compared to other programming languages like asp.net( is server side language)
* Java script is a client side language thus it does not need any assistance of web server to execute.
* Java script now is also server side language(node.js)

Advantages:

* Less server interaction
* Immediate feedback to the visitors
* Increase interactivity
* Richer interfaces

# Object in java script?

Object is an entity it having states and behaviors. Java script is object based scripting language.

3ways to create objects

* Object literal
  + Object ={ property1:value1,property2:value2,property3:value3}
  + Var emp = { name: ”ram”, id:101};
  + Document .write(emp.name +””+emp.id}
* Creates instance of object
  + Var emp = new emp()
  + Emp.id=101;
  + Emp.name = “ram “
  + Document .write(emp.id+””+emp.name)
* Using an object constructors
  + Function emp(id,name){
  + This.id=id
  + This.name= name;
  + }
  + E=new emp(102,”ram”)
  + Docment.write(emp.id+””+emp.name)

# Java script data types?

* primitive
  + String
  + Number
  + Boolean
  + Object
  + Undefined
* Non primitive
  + Array
  + Object
  + Regular expression

# What is the use of isNAN function?

NAN = not a number

IsNan function returns the true value, if the argument is not a number otherwise it is false value.

# What is prompt box?

Allow the user to enter the input by providing the text box.

# What is this keyword in java script?

This key refers to the object from where it was called

# What is variable typing in java script?

Variable typing is used to assign a number to variable and same variable can be assigned to string.

I = 10

I = “string”;

This is called variable typing.

# What do you mean by Null in java script?

The Null value is used to represent no value or no object.

# What is an undefined value in java script?

Undefined value means

Variable used in the code does not exist.

Variable is not assigned to any value.

Property does not exist.

# Types of popup box available in java script?

* Alert 🡪 it shows only ok button
* Confirm 🡪 it shows two buttons like confirm and cancel.
* Prompt

# What is break and continue statement

* Break statement exits from the current loop.
* Continue statement continues with next statement of the loop

# Which keywords are used to handle the exceptions?

* Try catch and finally is used to handle exceptions in java script.

# Types of errors in java script?

* Run time errors:
  + Misuse of commands inside the html language.
* Logical error:
  + When we perform the bad logic in the function.
* Load time errors:
  + Loading a web page like improper syntax error is known as logical error.
  + It generates the errors dynamically.

# What are the scope of a variable in java script?

Java script variable will have two scopes

* Global variable:
  + Global variable has global scope .it is visible everywhere in your java script code.
* Local variable :
  + It is visible only with in a function where it is defined.

# What is callback?

It is a function that is to be executed after another function has finished executing.

Function to be executed one after another function finished executing.

# Difference between attribute and property

* Attribute
  + Provide more details on an element like id type value.
* Property
  + Value assigned to the property like type =”text” value =”name”.

# Difference between var let const

* Variable
  + Variable allows duplicates
  + Function scope
* Let
  + Block scope
  + We can change the value of variable any number of times
  + {  
    let first\_num =1;  
    first\_num=2;  
    document. write (first\_num);  
    }
* Const
  + After the first assignment of value we cannot redefine the value again
  + Does not change the values.
  + {  
    const second\_num =1;  
    second\_num=2;  
    document. write (second\_num);  
    }

# What is typed language?

* Dynamically typed language:
  + Variable can hold multiple values.
* Statically typed language:
  + Variable can hold single value

# Local and session storage?

* Local storage
  + Data is not sent back to the server for every http request.
  + Reduce the amount of traffic between the client and server.
* Session storage:
  + When data is stored in the local storage it has no expiration date.
  + When data is stored in the session storage, data is expired when the page session ends.

# Strict mode in java script?

It is a Way to introduce better error-checking into your code.

# What is hosting?

If you have variable declared anywhere inside the code then it is bought up to the top

This methods is only applicable for declaration of variable and is not applicable for the initialization variable;

Functions are hosted to top.