1. **what is angular ?**

Angular is type script based framework mainly used to create spa

* **component**: Components define *views*, which are sets of screen elements that Angular can modify according to your program logic and data.

1. **what is angular services?**

* Angular services are singleton objects which mainly instantiated once during life cycle.
* And also used to share all business logic across all component and directive.
* A service class definition is immediately preceded by the @[Injectable](https://angular.io/api/core/Injectable)() decorator.
* The decorator provides the metadata that allows other providers to be **injected** as dependencies into your class.

1. **Advantages of Angular:**

* custom directive
* client and server communication
* community support
* MVC pattern
* Two way data binding
* Dependency injection

1. **Angular VS Angular JS**

* Angular js is based on JavaScript
* Angular js is not faster than angular2
* Angular js has a concept of controller
* components are the building blocks of angular2. and view.

1. **Angular Features:**

* Accessibility application
* cross platform development
* Animation support
* Template
* Testing using protractor

1. **IVY Engine And Bezel Engine:**

* **Ivy Engine:**- it reduces the bundle size which increase its performance and works faster in slow network.
* **Bezel Engine**:- it has smart algorithm that keeps track of build dependencies
* **Differential Loading**:- it chooses b/w modern and legacy java script as per their capabilities. when user load application they will get the bundle they need.

1. **Authentication VS Authorization:**

* **Authentication**:- when user login, user login credentials send to authenticated api that returns jwt(JSON web token) in response that contains user information for authenticating user.
* **Authorization**:- post login roles given to user what he can access or not.

1. **Templates**:

Templates are written by angular with specific elements and attributes. it mainly used to display dynamic view rendered from component.

1. **Annotation Vs Decorator:-**

* Annotation is used to create annotation array and it's a metadata set of class of library.
* Decorators are used to decorate and modify class without altering source code

1. **Directive:**

* **Component Directive**: directive with template.
* **Structural Directive**: changing the Dom layout by adding and removing Dom element.
* **Attribute Directive**: changing the appearances and behavior or component or other directive

1. **Attribute ([colSapn])vs. property([disabled]):**

* Attribute value can't change, prop value can change.
* Attribute binds by html tags and prop bind by Dom.

1. **Types Of Binding:**

* Interpolation
* Property Binding
* Two way binding
* Event Binding

1. **Angular Vs Jquery:**

* Jquery doesn't support restful api.
* Jquery doesn't support two way binding.
* Jquery doesn't support routing.

1. **Building Blocks:**

* **component**: it's a building blocks of angular defined to be visible in each view.
* **Data Binding**: its mainly used to connect both component and template.
* **Dependency Injection**: its mainly use to inject the instance of class.
* Directive
* **Metadata:** it tells the angular that it is a component not a class.
* **Modules** : it's a group of component ,directive and services.
* **Routing**
* **Services**
* **Template**

1. **Angular Material:**

Angular material is a ui component which is mainly used to design attractive and functional web pages like device independence, browser compatibility.

1. **Angular Life Cycle Hooks:**

* **Constructor**: it invoked when component is created
* **ngOnChanges**: it invoked if any changes detected.
* **ngOnInit**: it invoked when component initialized .it invoked once during whole life cycle.
* **ngDoCheck**: it performs when any change detection occurs.
* **ngonDestroy**: it performs just before any component destroyed.

1. **View Encapsulation:**

* **Emulated**: style defined in component spread to others.
* **Native**: style not defined in component spread to others.
* **None**: style defined in component visible to all.

1. **DOM VS BOM:**

* Document object model used to represent content of web page.
* Bom is browser object object model used to add browser attributes.

1. **Pipes**

it takes input and transform it to desired output.

1. **Observable:**

* Angular uses observer pattern. Observable objects are registered and other objects observe them using subscribe method when they acted in some way.
* observable can be cancelled.
* observable retry on error.
* observable are lazy they will not call until subscribed.
* it emits multiple value.

1. **Promises:**

* it can't be cancelled.
* it is not lazy.
* it emit single value.

1. **RxJs:**

it's a library consist of asynchronous and call back function in a functional and reactive way using observables.

ex. Http Client produces and consumes observables.

1. **AOT:**

* it's a ahead of time compiler.
* it coverts angular and html code to JavaScript during the build phase. before browser download in runs it.
* compiling your application during build process provides faster rendering in browser.

1. **Web Workers:**

it performs large intensive cpu computations and freeing main thread to update user interface.

1. **Difference b/w js and ts:**

**JS:**

* it provide .js extension.
* it is scripting language.
* it gives error at runtime.
* it directly run into the browser.
* it doesn't support modules.