Component:

Angular is **an open-source JavaScript framework for front end development of web applications**

Components are **the main building block for Angular** applications. Each component consists of: An HTML template that declares what renders on the page. A Typescript class that defines behavior. A CSS selector that defines how the component is used in a template

What is the interpolation in angular?

**Interpolation:** *It represents as {{}}. it may be concatenate two string ,calculate value and display value.*

**Property Binding:** *It represents as [].It is mainly used for non-concatenate string like variable*

How many ways b/w the components.?

* Parent to child component
* Child to parent component
* Sharing data between sibling components
* Sharing data using ViewChild property
* Sharing data between not related components

The child component uses **the @Output() property to raise an event to notify the parent of the change**. To raise an event, an @Output() must have the type of EventEmitter , which is a class in @angular/core that you use to emit custom events.

Parent to child:  
A common pattern in Angular is **sharing data between a parent component** and one or more child components. To implement this pattern use the @Input()

What is the Authentication and authorization to ?

Authentication is the process matching the visitor of a web application with the pre-defined set of user identity in the system. ... Authorization is **the process of giving permission to the user to access certain resource in the system**.

What is the lazy loading concepts?

Lazy loading is a technique in Angular that **allows you to load JavaScript components asynchronously when a specific route is activated**. It improves the speed of the application load time by splitting the application into several bundles. When the user navigates through the app, the bundles are loaded as required

How many ways inject the services?

There are three types of dependency injection — **constructor injection, method injection, and property injection**

@NgModule() or @Injectable() annotation.

Use the **@Injectable() decorator** on any service that depends on another. Inject the other services into the constructor of the dependent service.

Use of services?

A reusable Angular service is **designed to encapsulate business logic and data with different components of Angular**.

**@Inject()**

The Inject is a parameter decorator on a dependency parameter of a class constructor that specifies a custom provider of the dependency

@Injectable()

The @Injectable decorator as a class provided and Injected as a dependency.

What is Providers ?

The service itself is a class that the CLI generated and that's decorated with @Injectable() . By default, this decorator has a providedIn property, which creates a provider for the service. In this case, providedIn: 'root' specifies that Angular should provide the service in the root injector.

What is Custome pipes ?

 Custom Pipes can be of two types. Pure and impure custom pipe. Pipe **takes an input and returns an output based on** the output of transform function evaluation. Usage of Pipes. Angular has many built-in pipes

What is the directives?

Directives with templates known as **Components**. **Directives** that creates and destroys DOM elements known as Structural Directives. Directives that manipulate DOM by changing behavior and appearance known as Attribute Directives

Custome directives ?

Custom directives are used in **AngularJS to extend the functionality of HTML**. Custom directives are defined using "directive" function. A custom directive simply replaces the element for which it is activated

What is the Rxjs concepts?

RxJS is an awesome library when it comes to the handling of async tasks. RxJS uses observables to work with reactive programming that deals with asynchronous data calls, callbacks and event-based programs.

Map?

RxJS map() operator is **a transformation operator used to transform the items emitted by an Observable by applying a function to each item**. It applies a given project function to each value emitted by the source Observable and then emits the resulting values as an Observable.

Filter?

RxJS filter() operator is **a filtering operator used to filter items emitted by the source observable according to the predicate function**. ... This operator takes values from the source Observable, passes them through a predicate function and only emits those values that get TRUE

Reduce?

The RxJS reduce() operator is a mathematical operator that **applies an accumulator function over the input or source Observable**, which returns an accumulated value in an observable form when the source is completed. It also gives an optional seed value.

What is the difference between behavior subject and subject?

A BehaviorSubject holds one value. When it is subscribed it emits the value immediately. **A Subject doesn't hold a value**

**What is the Angular zone?**

NgZone gives you back control of your code's execution. There are two relevant methods in NgZone – **run and runOutsideAngular**: runOutsideAngular runs a given function outside the angular zone, meaning its code won't trigger change detection. run runs a given function inside the angular zone.

**What is the folder structure?**

Inside the **src/ folder**, the app/ folder contains your project's logic and data. Angular components, templates, and styles go here. Defines the logic for the application's root component, named AppComponent .

**What is the structure of the loading process?**

**First load the angular.json , main.ts module.ts app.component.ts index.html and app.component.html**

**What is the life cycle in angular?**

1)**Constructor**: A default method which is called when the class is instantiated.

2)**ngOnChanges**: Executes when a new component is created, when one of the bound properties with @Input changes, also it is the only hook that takes an argument when it is called which is called as SimpleChanges.

3)**ngOnInit**: Called once the component is initialized. This doesn't allow the component to be visible over the DOM. This runs just after the constructor.

4)**ngDoCheck**: Runs when change detection runs. It also runs if there is no change and even if it is just an event that occurred, just in case to make sure if there is something that has changed. (for eg: It will run after a button click event irrespective of that it is making ant change or not) 5)**ngAfterContentInit**: This is called after content(ng-content) has been projected into the view.

6)**ngAfterContentChecked**: This is called after every projected content has been checked.

7)**ngAfterViewInit**: Called after the components view (and child view) has been initialized.

8)**ngAfterViewChecked**: Called every time the view (and child view) has been checked.

9) **ngOnDestroy**: Called when we generally use an if condition and render the component accordingly. This is mainly called right before the object is destroyed by the angular.

What is the ngIf and ngHide?

The ng-hide directive shows or hides the given HTML element based on the expression provided to the ng-hide attribute . ng-if can only render data whenever the condition is true. ... ng-show **can show and hide the rendered data**, that is, it always kept the rendered data and show or hide on the basis of that directives.

What is the interceptor?

Interceptors are a unique type of Angular Service that we can implement. Interceptors **allow us to intercept incoming or outgoing HTTP requests using the HttpClient** . By intercepting the HTTP request, we can modify or change the value of the request.

A Resolver is **a class that implements the Resolve interface of Angular Router**. In fact, Resolver is a service that has to be [provided] in the root module. Basically, a Resolver acts like middleware, which can be executed before a component is loaded.