1. What is Angular?

Angular is a TypeScript-based open-source web application framework developed by Google. It is used for building single-page applications (SPAs) by extending HTML with directives and binding data efficiently.

2. What are the key features of Angular?

Key features include two-way data binding, dependency injection, directives, components, services, and a modular architecture for building dynamic and scalable web applications.

3. What is a Component in Angular?

A component is the core building block of an Angular application, consisting of an HTML template, CSS styles, and TypeScript code. It controls a part of the user interface and handles user interactions.

4. What is a Module in Angular?

A module is a container for different parts of an application, such as components, directives, services, and pipes. It helps in organizing the application into cohesive blocks and supports lazy loading.

5. What is Data Binding in Angular?

Data binding refers to the synchronization of data between the model (component) and the view (template). Angular supports four types: interpolation, property binding, event binding, and two-way binding.

6. What is Dependency Injection in Angular?

Dependency Injection (DI) is a design pattern used in Angular to provide instances of services and components. It helps in managing dependencies, making code more modular and testable.

7. What is a Directive in Angular?

A directive is a special marker in the DOM that tells Angular to do something with a DOM element. It can be an attribute, structural, or a custom directive to manipulate the behavior of elements.

8. What is the purpose of Services in Angular?

Services in Angular are classes that encapsulate reusable business logic and data access operations. They can be injected into components or other services using dependency injection.

9. What is the Angular CLI?

Angular CLI (Command Line Interface) is a powerful tool that automates common development tasks, such as creating components, services, and building the application. It helps in maintaining a consistent structure.

10. What is a Pipe in Angular?

A pipe is a function that transforms data in Angular templates. It can be used to format, filter, or modify data displayed in the view, such as converting dates or changing text case.

11. What is an Angular Template?

A template in Angular defines the view of a component. It is written in HTML and can include Angular syntax like interpolation, directives, and binding expressions to dynamically render content.

12. What is Routing in Angular?

Routing in Angular allows navigation between different views or pages within a single-page application. The Angular Router module manages this, enabling users to move between different components.

13. What is Lazy Loading in Angular?

Lazy loading is a technique in Angular where modules are loaded on demand rather than at the initial load. This helps in reducing the application's load time and improves performance.

14. What is Angular's Change Detection?

Change detection in Angular is a mechanism that automatically updates the view whenever the data model changes. Angular provides both default and onPush strategies for change detection.

15. What is Two-Way Data Binding in Angular?

Two-way data binding allows synchronization of data between the view and the component. Changes made in the view update the component's model and vice versa, using the `ngModel` directive.

16. What is the difference between `ngIf` and `ngSwitch`?

`ngIf` conditionally adds or removes an element from the DOM based on a boolean expression, whereas `ngSwitch` allows switching between multiple elements based on a matching expression.

17. What is an Angular Lifecycle Hook?

Lifecycle hooks are special methods in Angular that allow you to tap into key moments of a component's life, such as creation, update, and destruction. Examples include `ngOnInit` and `ngOnDestroy`.

18. What is the difference between `constructor` and `ngOnInit`?

The `constructor` is called when an instance of a class is created, mainly used for dependency injection. `ngOnInit` is called after the constructor and is used for initialization logic.

19. What is the purpose of `ngOnDestroy`?

`ngOnDestroy` is a lifecycle hook called before a component or directive is destroyed. It is used for cleanup tasks like unsubscribing from observables or detaching event handlers.

20. What is AOT compilation in Angular?

AOT (Ahead-of-Time) compilation compiles Angular templates and components during the build process, resulting in faster rendering and smaller bundle sizes at runtime.

21. What is the difference between Template-Driven and Reactive Forms?

Template-driven forms rely on Angular's directives in the template for form validation and handling, whereas reactive forms use a more programmatic approach, providing more control and flexibility.

22. What is `ngModel` in Angular?

`ngModel` is a directive in Angular that creates a two-way data binding between the form input and the component model, commonly used in template-driven forms.

23. What is the difference between `@Component` and `@Directive` in Angular?

`@Component` is a decorator for creating a component, which controls a part of the user interface. `@Directive` is used for creating custom directives that manipulate the DOM without a template.

24. What is Angular's HttpClient?

HttpClient is an Angular service used for making HTTP requests to communicate with backend services. It supports handling of requests, responses, error handling, and observables.

25. What is `RxJS` in Angular?

RxJS (Reactive Extensions for JavaScript) is a library for composing asynchronous and event-based programs using observables. It is heavily used in Angular for handling asynchronous operations like HTTP calls.

26. What is an Observable in Angular?

An Observable is a core concept in RxJS that represents a stream of data that can be observed over time. Angular uses Observables for handling asynchronous operations like HTTP requests and event handling.

27. What is the difference between `Observable` and `Promise` in Angular?

An Observable is more powerful than a Promise, as it can handle multiple values over time, support cancellation, and offer operators for handling data streams, whereas a Promise resolves to a single value.

28. What is Angular Universal?

Angular Universal is a server-side rendering (SSR) technology for Angular applications. It renders the application on the server before sending it to the client, improving performance and SEO.

29. What is the purpose of `ngFor` in Angular?

`ngFor` is a structural directive used to iterate over a collection and render an element for each item. It simplifies displaying lists in Angular templates.

30. What is the difference between `ngClass` and `ngStyle`?

`ngClass` allows you to dynamically add or remove CSS classes to an element based on conditions. `ngStyle` enables you to set CSS styles dynamically based on component properties.

31. What is a Guard in Angular?

Guards are used to control access to routes in Angular applications. They can prevent users from accessing certain routes or handle authorization checks before navigation.

32. What is a Resolver in Angular?

A Resolver is used to pre-fetch data before navigating to a route. It ensures that the required data is available for the component before it loads.

33. What is Angular's `Zone.js`?

`Zone.js` is a library used in Angular for tracking asynchronous operations and triggering change detection automatically when those operations complete.

34. What is Angular Material?

Angular Material is a UI component library that follows Google's Material Design principles. It provides pre-built, reusable components for building responsive and modern Angular applications.

35. What is Angular's Ivy?

Ivy is the latest rendering engine in Angular that improves performance, reduces bundle size, and offers better debugging and compilation features compared to the previous renderer.

36. What is the purpose of `trackBy` in `ngFor`?

`trackBy` is a function used in `ngFor` to improve performance by tracking items in a list by their unique identifier, preventing unnecessary re-renders when items change.

37. What is Angular's Service Worker?

Angular's Service Worker is a script that runs in the background and enables features like offline caching, background data synchronization, and push notifications in Angular applications.

38. What is the purpose of `ng-content`?

`ng-content` is used to project content from a parent component into a child component's template. It allows creating reusable components with customizable content.

39. What is the purpose of `ng-template`?

`ng-template` is an Angular directive used to define template fragments that are rendered dynamically. It is often used with structural directives like `ngIf` and `ngFor`.

40. What is an Angular Router Outlet?

A Router Outlet is a directive used in Angular to display components based on the active route. It serves as a placeholder for dynamic content controlled by the router.

41. What is Angular's `ViewChild`?

`ViewChild` is a decorator in Angular that allows you to access a child component, directive, or DOM

element from the parent component. It is commonly used for manipulating DOM elements programmatically.

42. What is Angular's `ngZone`?

`ngZone` is a service that allows Angular to manage change detection efficiently. It helps in executing code inside or outside Angular's zone to control when change detection should run.

43. What is Angular's TemplateRef?

TemplateRef represents an Angular template, which can be used to reference and instantiate embedded views. It is commonly used with `ng-template` for dynamic view creation.

44. What is the difference between `ViewChild` and `ContentChild`?

`ViewChild` accesses elements or components within the component's view, while `ContentChild` accesses projected content, such as content passed from a parent component.

45. What is Angular's `Renderer2`?

`Renderer2` is an Angular service used for safely manipulating DOM elements, attributes, and styles. It provides an abstraction layer that ensures compatibility across different platforms.

46. What is Angular's `ActivatedRoute`?

`ActivatedRoute` is a service that provides access to information about the route associated with the component, such as route parameters, query parameters, and URL fragments.

47. What is Angular's `Location` service?

The `Location` service provides an API for interacting with the browser's URL and history stack. It allows for programmatic navigation and URL manipulation in Angular applications.

48. What is Angular's `RouterLink`?

`RouterLink` is a directive that enables navigation between routes declaratively in templates. It is used to link to different routes without triggering a full page reload.

49. What is the purpose of `ElementRef` in Angular?

`ElementRef` is a wrapper around a native DOM element that allows direct access to the element in the component. It is used for low-level DOM manipulation.

50. What is the purpose of `ngOnChanges`?

`ngOnChanges` is a lifecycle hook that is called when an input property of a component changes. It is used to detect and respond to changes in input data.

For Experienced :

1. What is Angular Ivy?

Ivy is Angular's rendering engine, improving build speed, bundle size, and debugging.

2. How does Angular handle state management?

Angular can handle state using services, NgRx, or other state management libraries like Akita.

3. What are the benefits of using Angular Universal?

Angular Universal enables server-side rendering (SSR), improving SEO and initial load performance.

4. Explain the role of `ngZone` in Angular.

`ngZone` manages Angular's change detection by controlling the execution of code inside or outside Angular's zone.

5. What is the purpose of `forwardRef` in Angular?

`forwardRef` allows circular dependency resolution, deferring the reference of a provider until it's defined.

6. How does Angular implement lazy loading?

Angular uses the `loadChildren` property in the route configuration to load modules on demand.

7. What is differential loading in Angular?

Differential loading generates two bundles: one for modern browsers and one for legacy browsers.

8. How does Angular handle error handling in HTTP calls?

Angular uses HttpClient's `catchError` operator with RxJS to handle errors in HTTP requests.

9. Explain the concept of `dynamic components` in Angular.

Dynamic components are components that are created and inserted into the DOM at runtime using `ComponentFactoryResolver`.

10. What is `content projection` in Angular?

Content projection allows you to insert external content into a component using `<ng-content>`.

11. How do you optimize Angular applications for performance?

Optimize by using lazy loading, Ahead-of-Time (AOT) compilation, OnPush change detection, and minimizing bundle size.

12. What is Angular's Renderer2?

Renderer2 provides a platform-independent way to manipulate the DOM safely in Angular.

13. How does Angular handle forms?

Angular supports two types of forms: template-driven forms for simple use cases and reactive forms for more complex scenarios.

14. What is the difference between `mergeMap` and `switchMap` in RxJS?

`mergeMap` subscribes to all inner observables, while `switchMap` cancels the previous observable and subscribes to the latest one.

15. How does Angular's `@HostListener` work?

`@HostListener` decorates a method to listen for DOM events on the host element and trigger the method when the event occurs.

16. What is `Angular Elements`?

Angular Elements allows you to create custom elements (Web Components) from Angular components, making them reusable outside Angular.

17. Explain the `async pipe` in Angular.

The `async` pipe automatically subscribes to an Observable or Promise and handles the unsubscription when the component is destroyed.

18. What is the role of the `trackBy` function in `ngFor`?

The `trackBy` function helps Angular track items in a list by their unique identifier, improving rendering performance.

19. How do you implement route guards in Angular?

Implement route guards using `CanActivate`, `CanDeactivate`, and other interfaces to control access to routes.

20. Explain Angular's ViewEncapsulation.

ViewEncapsulation determines how styles are applied to components, using options like Emulated, None, or Shadow DOM.

21. What are Angular schematics?

Angular schematics are code generators that create or modify projects using Angular CLI commands.

22. What is the difference between `Reactive Forms` and `Template-Driven Forms`?

Reactive forms provide a model-driven approach with better control, while template-driven forms are simpler and use directives in the template.

23. How does Angular handle routing animations?

Angular uses the `Router` module in combination with Angular animations to animate route transitions.

24. What is Angular's `Service Worker`?

The Service Worker allows Angular applications to cache assets and API responses, enabling offline capabilities.

25. How do you handle component interaction in Angular?

Handle component interaction using Input/Output decorators, shared services, or `EventEmitter`.

26. What is the purpose of the `Injector` in Angular?

The `Injector` is responsible for instantiating components and services and managing their dependencies.

27. What is the role of Angular's `TestBed`?

`TestBed` is used for configuring and initializing an environment for unit testing Angular components and services.

28. How do you configure custom pipes in Angular?

Create custom pipes by implementing the `PipeTransform` interface and using the `@Pipe` decorator.

29. What is the difference between `@ViewChild` and `@ContentChild`?

`@ViewChild` queries elements within the component's template, while `@ContentChild` queries projected content from parent components.

30. How does Angular's `async/await` work with Observables?

Angular uses `async/await` to work with Promises; for Observables, you need to convert them to Promises or use RxJS operators.

31. What is Angular's `ActivatedRouteSnapshot`?

`ActivatedRouteSnapshot` provides the current state of the route at a particular moment, without tracking future changes.

32. What is the difference between `HttpClientModule` and `HttpModule`?

`HttpClientModule` is a more modern and feature-rich alternative to `HttpModule`, introduced in Angular 4.3.

33. Explain Angular's `Renderer2`.

`Renderer2` is a service that abstracts DOM manipulations to ensure compatibility across different platforms.

34. What is the difference between `constructor` and `ngOnInit` in Angular?

The constructor is for dependency injection, while `ngOnInit` is for component initialization logic.

35. What is Angular's `ng-template`?

`ng-template` defines an Angular template that is not rendered by default but can be conditionally rendered.

36. What is the use of `@Injectable` in Angular?

`@Injectable` marks a class as available to be provided and injected as a dependency within the Angular application.

37. How do you handle forms validation in Angular?

Angular handles form validation using built-in validators and custom validators for reactive and template-driven forms.

38. What is the role of the `Router` module in Angular?

The `Router` module manages navigation, URL handling, and route configuration in Angular applications.

39. What is the difference between `@Input` and `@Output` in Angular?

`@Input` passes data from parent to child components, while `@Output` emits events from child to parent components.

40. How do you implement `ngOnChanges` in Angular?

Implement `ngOnChanges` to detect and act on changes to input properties in a component.

41. What is Angular's `PreloadingStrategy`?

`PreloadingStrategy` determines how Angular preloads lazy-loaded modules after the application has finished loading.

42. Explain the `catchError` operator in Angular.

`catchError` is an RxJS operator that handles errors in observables and allows for error handling strategies.

43. How does Angular's `OnPush` change detection strategy work?

`OnPush` triggers change detection only when input properties change, improving performance.

44. What is Angular's `HttpInterceptor`?

`HttpInterceptor` intercepts HTTP requests and responses, allowing you to modify them before passing them along.

45. What is Angular's `Dependency Injection` hierarchy?

Angular manages dependencies using a hierarchical injector system, allowing for local and global service providers.

46. What is Angular's `ElementRef` used for?

`ElementRef` gives direct access to the native DOM element in Angular, typically used for low-level DOM manipulation.

47. What is the difference between `ActivatedRoute` and `RouterState`?

`ActivatedRoute` represents the active route associated with the component, while `RouterState` represents the entire state of the router at a specific moment.

48. What are Angular's custom decorators?

Angular allows creating custom decorators for reusable logic, similar to `@Input`, `@Output`, and others.

49. What is Angular's `ViewContainerRef`?

`ViewContainerRef` provides an API to manipulate elements inside a view container, allowing dynamic component creation and insertion.

50. What is the purpose of `Angular Schematics`?

Angular Schematics automates project setup and modifications, ensuring consistency and best practices across Angular applications.