Angular Interview Questions and Answers (Basic to Intermediate)

1. What is Angular?

Angular is a TypeScript-based front-end framework developed by Google to build dynamic, single-page web applications (SPAs).

Example: Applications like Gmail or Google Docs that load once and dynamically update without refreshing use Angular.

2. What are components in Angular?

Components control a part of the UI. Each component has:

- A TypeScript class
- An HTML template
- Optional CSS styles

Example: A shopping cart component displays selected items, handles quantity updates, and totals.

3. What is a module in Angular?

A module (`@NgModule`) organizes components, directives, and services. Every Angular app has at least one root module (`AppModule`).

Example: You can create a `UserModule` for login, registration, and profile components.

4. What is data binding in Angular?

It connects the component's data with the template:

- Interpolation: `{{ title }}`
- Property Binding: `[src]="imageUrl"`
- Event Binding: `(click)="handleClick()"`
- Two-way Binding: `[(ngModel)]="name"`

5. Difference between interpolation and property binding?

- Interpolation: Sets **text** content - `{{title}}`
- Property Binding: Binds **element properties** - `[value]="title"`

Example: `<img [src]="imageUrl">` vs. `<p>{{imageUrl}}</p>`

---

6. What is two-way data binding?

Synchronizes data between component and template using `[(ngModel)]`.

Example:

```
<input [(ngModel)]="username">  Hello {{username}}
```

---

7. What is a directive in Angular?

A directive modifies DOM behavior:

- Structural: e.g., `*ngIf`, `*ngFor`
- Attribute: e.g., `ngClass`, `ngStyle`

---

8. Difference between structural and attribute directives?

- Structural changes layout (`*ngIf`, `*ngFor`)
- Attribute changes appearance/behavior (`[ngClass]`, `[ngStyle]`)

---

9. What is Angular CLI?

A command-line tool to initialize, develop, and maintain Angular apps.

Example:

```
ng new my-app
ng generate component login
ng serve
```

---

10. What are services in Angular?

Services hold business logic and reusable code. Injected into components.

Example: `AuthService` handles login logic and can be used in multiple components.

---

11. What is dependency injection in Angular?

A design pattern where dependencies (like services) are injected instead of being created inside components.

Example:

```
constructor(private authService: AuthService) {}
```

---

12. What are lifecycle hooks?

Lifecycle hooks let you act during key moments in a component's life:

- `ngOnInit`, `ngOnChanges`, `ngOnDestroy`, etc.

---

13. Purpose of `ngOnInit()`?

Runs after component initialization. Commonly used for fetching data.

Example:

```
ngOnInit() {
  this.loadUserData();
}
```

---

14. What is routing in Angular?

Routing navigates between components using URL paths.

Example:

```
{ path: 'login', component: LoginComponent }
```

---

15. Use of `RouterModule`?

Enables routing configuration in an Angular app using `RouterModule.forRoot(routes)`.

---

16. What is a pipe in Angular?

Pipes transform data in templates (e.g., `date`, `uppercase`, `currency`).

Example: ``` {{ ``price`` | currency:'INR' }} ```

---

17. Difference between pure and impure pipes?

- Pure pipes run only when input changes.
- Impure pipes run on every change detection.

---

18. What is `ngIf` and `ngFor`?

- `*ngIf`: conditionally includes a block
- `*ngFor`: loops over a list

Example:

```
<div *ngIf="isLoggedIn">Welcome!</div>
<li *ngFor="let item of items">{{ item }}</li>
```

---

19. What is the role of `@Input()` and `@Output()`?

- `@Input()`: passes data from parent to child
- `@Output()`: emits event from child to parent

---

20. What are observables in Angular?

Used for asynchronous data streams (e.g., HTTP requests).

Example:

```
this.http.get('url').subscribe(data => console.log(data));
```

---

21. What is lazy loading in Angular?

Loading feature modules only when needed to improve performance.

Example:

```
loadChildren: () => import('./admin/admin.module').then(m => m.AdminModule)
```

---

22. Difference between template-driven and reactive forms?

- Template-driven: simple, uses `ngModel`, suitable for small forms.
- Reactive: complex, uses `FormGroup`, scalable.

---

23. How do you handle form validation?

- Template: using `required`, `pattern`, etc.
- Reactive: using `Validators.required`, etc.

---

24. What is a guard in Angular?

Guards control access to routes.

Example: Prevent unauthorized access to dashboard.

---

25. Types of guards?

- `CanActivate`
- `CanDeactivate`
- `Resolve`
- `CanLoad`

---

26. What is change detection?

Angular’s mechanism to update the view when the model changes.

---

27. Difference between `ngOnChanges()` and `ngDoCheck()`?

- `ngOnChanges()`: called on `@Input()` value change
- `ngDoCheck()`: called on every change detection run

---

28. How does Angular detect and update the DOM?

It uses zone.js to listen to async events and runs change detection to update the DOM accordingly.

---

29. What is the `async` pipe?

Automatically subscribes and unsubscribes from Observables in templates.

Example: `{{ user$ | async }}`

---

30. What is a resolver?

Pre-fetches data before a route is activated.

---

31. How does Angular handle HTTP requests?

Using `HttpClient` service from `HttpClientModule`.

---

32. What is `HttpClientModule`?

Provides tools to make HTTP requests, parse responses, and handle errors.

---

33. Difference between `subscribe()` and `async` pipe?

- `subscribe()`: manually subscribe to Observable
- `async` pipe: auto-subscribe in template

---

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

Used to project content into a component.

Example:

```
<ng-content></ng-content>
```

---

35. What is a custom directive?

A user-defined directive to extend behavior.

Example:

```
@Directive({ selector: '[appHighlight]' })
```

---

36. What is a shared module?

A module to share commonly used components, pipes, and directives.

---

37. What is a feature module?

Used to group related functionality (e.g., AdminModule, UserModule).

---

38. What is AOT compilation?

Angular compiles the app during build time (not runtime). Improves performance.

---

39. Difference between `providedIn: 'root'` and `providers[]`?

- `'root'`: Singleton service app-wide
- `providers[]`: Service is scoped to that module/component

---

40. What is a standalone component?

Angular 14+ allows creating components without declaring them in a module.

Example:

```
@Component({ standalone: true })
```