🔹 Core TypeScript Knowledge

1. What are the main differences between **TypeScript and JavaScript**? Why is TypeScript preferred in Angular projects?
2. What is the difference between **type aliases** and **interfaces**? When would you choose one over the other?
3. Explain the difference between any, unknown, never, and void.
4. What is **strict mode** in TypeScript? How does it impact Angular development?
5. How does **structural typing** work in TypeScript?

🔹 Generics & Advanced Types

6. Can you explain **generics** in TypeScript with an example?
7. What is the difference between <T extends object> vs <T extends keyof object>?
8. How do you use **conditional types** (extends ? :) in TypeScript?
9. What are **mapped types** in TypeScript? Can you give an example using Partial, Pick, or Record?
10. What is the purpose of **utility types** like Omit, Readonly, and Exclude?

🔹 Decorators & Angular Context

11. How do TypeScript **decorators** work under the hood?
12. Angular heavily uses decorators (@Component, @Injectable). Can you explain how TypeScript metadata (reflect-metadata) ties into Angular's DI system?
13. How would you write a **custom class/property decorator** in TypeScript?

🔹 Type Inference & Narrowing

14. How does **type inference** work in TypeScript?
15. What is **type narrowing**? Can you give an example using typeof, instanceof, or custom type guards?
16. Explain how **discriminated unions** work in TypeScript and why they're useful.
17. What is the difference between in operator narrowing vs instanceof narrowing?

🔹 **Asynchronous Type Safety**

18. How do you strongly type an async function's return value?
19. How do you type an **HTTP call** in Angular using HttpClient<T> with generics?
20. How do you type a function that returns either Promise<T> or Observable<T>?

🔹 **Real-World Scenarios**

21. You have a large interface with 50+ properties but you only need 3 in a component — how would you type it efficiently?
22. You're creating a reusable Angular form component — how do you strongly type the form model using FormGroup with generics?
23. How would you design a type-safe Angular service that returns API responses with pagination (data + meta)?
24. How do you enforce immutability in TypeScript for application state (e.g., NgRx)?
25. Have you ever written **type-safe RxJS operators** with generics?

🔹 **Best Practices & Pitfalls**

26. What's the difference between interface Foo {} and class Foo {} in TypeScript?
27. How do you avoid **excess property checks** in TypeScript when working with Angular inputs?
28. Why should you avoid any in Angular apps? What safer alternatives exist?
29. What's the difference between unknown and any in terms of type safety?
30. What's your approach to designing **scalable types** in a large Angular project?

✅ For **senior-level interviews**, expect follow-up coding exercises like:

- Writing a **generic Angular service** that fetches strongly typed API responses.
- Creating a **custom type guard** for a union type.
- Using Partial<T> or Omit<T> in a real-world Angular form scenario.