**📅 TypeScript Learning & Revision Plan (4 Weeks)**

💡 **Plan:**

* **Week 1-2:** Learn & Implement
* **Week 3:** Revision & Hands-on Practice
* **Week 4:** Real-World Projects & Interview Prep

**📌 Week 1: TypeScript Basics & Core Concepts**

**✅ Goal:** Understand the fundamentals of TypeScript and apply them in small programs.

**Day 1: Introduction & Setup**

🔲 What is TypeScript, and why use it?  
🔲 Install TypeScript (npm install -g typescript)  
🔲 Configure tsconfig.json  
🔲 Compile .ts files using tsc

**Day 2: Variables & Data Types**

🔲 var, let, const differences  
🔲 Primitive Types: string, number, boolean, bigint, symbol, null, undefined  
🔲 Object Types ({} syntax)  
🔲 Arrays (number[], Array<string>)  
🔲 Tuples ([string, number])  
🔲 Enums (enum Direction { Up, Down })

**Day 3: Functions**

🔲 Function Types and Signatures  
🔲 Optional & Default Parameters  
🔲 Rest Parameters (...args)  
🔲 Arrow Functions (const add = (a: number, b: number) => a + b)  
🔲 Function Overloading

**Day 4: Object-Oriented Programming (OOP)**

🔲 Classes (class, constructor, public, private, protected)  
🔲 Readonly properties (readonly)  
🔲 Getters and Setters  
🔲 Inheritance (extends, super())  
🔲 Abstract Classes (abstract class)  
🔲 Method Overriding  
🔲 Static Properties & Methods

**Day 5: Interfaces & Type Aliases**

🔲 Define an interface (interface Person { name: string; age: number; })  
🔲 Difference between interface and type  
🔲 Extend Interfaces (interface Employee extends Person { role: string; })  
🔲 Readonly properties in interfaces

**Day 6: Advanced Types**

🔲 Type Unions (string | number)  
🔲 Type Intersections (type A & type B)  
🔲 Literal Types (const status: "success" | "error")  
🔲 Mapped Types (type Optional<T> = { [K in keyof T]?: T[K] })  
🔲 Utility Types (Partial<T>, Readonly<T>, Pick<T>, Omit<T>)  
🔲 Index Signatures ({ [key: string]: number })

**Day 7: Practice & Recap**

🔲 Solve TypeScript exercises on variables, functions, and interfaces  
🔲 Implement a small program using **OOP concepts in TypeScript**

**📌 Week 2: Advanced TypeScript & Error Handling**

**✅ Goal:** Learn advanced TypeScript features and apply them in real-world scenarios.

**Day 8: Generics**

🔲 Generic Functions (function identity<T>(arg: T): T { return arg; })  
🔲 Generic Interfaces and Classes  
🔲 Generic Constraints (extends)  
🔲 Using Generics with Arrays and Objects

**Day 9: Modules & Namespaces**

🔲 import & export (default and named exports)  
🔲 Module Resolution  
🔲 Type Declarations (.d.ts files)  
🔲 Namespaces (namespace MyNamespace {})

**Day 10: Error Handling**

🔲 Try-Catch in TypeScript  
🔲 Custom Error Classes  
🔲 Handling Errors in Promises (catch())  
🔲 Error Handling in Async/Await  
🔲 never Type for Errors

**Day 11: Asynchronous TypeScript**

🔲 Callbacks in TypeScript  
🔲 Promises (Promise<T>)  
🔲 Async/Await (async function fetchData(): Promise<Data> {})  
🔲 Handling Async Errors (try-catch)

**Day 12: Type Guards & Advanced Operators**

🔲 Type Guards (typeof, instanceof, in)  
🔲 keyof Operator  
🔲 Conditional Types (T extends U ? X : Y)

**Day 13: Working with APIs in TypeScript**

🔲 Fetch API & Type Safety (fetch, Axios)  
🔲 Handling API Responses with Interfaces  
🔲 Strong Typing in Services and Components

**Day 14: Hands-on Mini Project**

🔲 Build a **small project using TypeScript**, such as a **To-Do App with strong typing**.

**📌 Week 3: Revision & Hands-on Practice**

**✅ Goal:** Revise all concepts, apply them to real-world projects, and practice interview questions.

**Day 15-16: Recap and Debugging**

🔲 Review **TypeScript Basics, Functions, OOP**  
🔲 Debug TypeScript code (sourceMap, DevTools)  
🔲 Fix TypeScript compilation errors (tsc --watch)

**Day 17-18: Real-World Scenarios**

🔲 Implement a **real-world API integration** using Axios and TypeScript  
🔲 Create a **modular TypeScript application** using import/export

**Day 19-20: Solve TypeScript Challenges**

🔲 Solve **TypeScript exercises** on interfaces, generics, and utility types  
🔲 Work on **real-world interview problems**

**📌 Week 4: Project Work & Interview Prep**

**✅ Goal:** Work on real-world applications, prepare for interviews, and practice coding questions.

**Day 21-22: Build a TypeScript Project**

🔲 Choose a **real-world project** (e.g., Task Manager, Weather App, Form Validator)  
🔲 Use **strong typing** and best practices

**Day 23-24: TypeScript in Angular**

🔲 Understand **Type Safety in Angular Services & Components**  
🔲 Learn **RxJS with TypeScript (Observable<T>)**

**Day 25-26: Prepare TypeScript Interview Questions**

🔲 Common Interview Questions (what is TypeScript, difference between interface and type)  
🔲 Advanced Questions (generics, utility types, async/await)

**Day 27-28: Mock Interviews & Final Revision**

🔲 Take **mock interviews** on TypeScript  
🔲 Revise important concepts **quickly using notes**

**📝 Final Checklist for TypeScript Mastery**

🔲 Do I understand **all TypeScript concepts**?  
🔲 Can I **write TypeScript code without errors**?  
🔲 Am I **comfortable using TypeScript in real-world projects**?  
🔲 Have I practiced **common TypeScript interview questions**?  
🔲 Can I **confidently explain concepts like generics, modules, and error handling**?

**🎯 What’s Next?**

Once you complete this **TypeScript roadmap**, you’ll be ready for:  
✅ **Advanced TypeScript-based Angular development**  
✅ **Real-world projects with TypeScript**  
✅ **TypeScript-related job interviews**