**🟢 FOUNDATION LEVEL**

✅ Chapter 1: What is TypeScript?  
✅ Chapter 2: tsconfig.json Explained (rootDir, outDir, strict)  
✅ Chapter 3: Basic Types — string, number, boolean, void, undefined, null  
✅ Chapter 4: Functions in TS — params, return, callback, overloads  
✅ Chapter 5: Special Types — any, unknown, never  
✅ Chapter 6: Union, Intersection, Literal Types  
✅ Chapter 7: Type vs Interface  
✅ Chapter 8: Tuples  
✅ Chapter 8.1: Tuple Deep Dive (key-value, named, optional, spread) 🆕  
✅ Chapter 9: Enums 🆕

**🟡 ADVANCED TYPE SYSTEM**

🔜 Chapter 10: Discriminated Unions  
🔜 Chapter 11: Type Guards & Exhaustiveness  
🔜 Chapter 12: keyof, typeof, indexed access  
🔜 Chapter 13: Mapped Types  
🔜 Chapter 14: Utility Types (Partial, Pick, Omit, Record…)  
🔜 Chapter 15: Conditional Types (T extends U ? X : Y)  
🔜 Chapter 16: Type Inference Deep Dive (infer, contextual typing)

**🔵 GENERICS & FUNCTIONAL TYPES**

🔜 Chapter 17: Generics  
🔜 Chapter 18: Generic Constraints  
🔜 Chapter 19: Built-in Generics (Promise<T>, ReturnType<T>)  
🔜 Chapter 20: Generic Functions & Components

**🟣 REAL-WORLD APPLICATIONS**

🔜 Chapter 21: TypeScript with React (Props, State, Context)  
🔜 Chapter 22: TypeScript with Redux Toolkit  
🔜 Chapter 23: TypeScript with Node.js + Express  
🔜 Chapter 24: Writing .d.ts Declaration Files  
🔜 Chapter 25: Module Resolution, tsconfig paths

**🧠 INTERVIEW, DEBUGGING & PRACTICE**

🔜 Chapter 26: Output-Based Questions 🔥  
🔜 Chapter 27: TypeScript Traps (any vs unknown, overloads, etc.)  
🔜 Chapter 28: Error Debugging Patterns  
🔜 Chapter 29: Type Challenges + Practice  
🔜 Chapter 30: Master Cheatsheet & Recap

**✅ Real-World Patterns**

* Typing API responses
* Typing functions callbacks & promises
* Working with async/await + TypeScript
* Narrowing API responses (success/error)
* Type-safe object and array patterns
* Working with enums & const assertions
* Readonly & immutable patterns

**React + TypeScript**

* Typing props, children, events
* useState, useRef, useReducer types
* Custom hooks typing
* Component generics
* Context API typing
* Redux Toolkit + TypeScript