TypeScript Syllabus

Here's a **complete TypeScript syllabus** from **basic to advanced**, structured in progressive learning stages:

**🧑‍💻 1. Introduction to TypeScript**

* What is [TypeScript](https://chatgpt.com?q=TypeScript)?
* Differences between [JavaScript](https://chatgpt.com?q=JavaScript) and TypeScript
* Benefits of using TypeScript
* Installing TypeScript (npm install -g typescript)
* Setting up a TypeScript project
* Compiling .ts files to .js (tsc command)
* tsconfig.json explained

**🔡 2. Basic Types**

* number, string, boolean
* any, unknown, void, null, undefined, never
* Arrays (number[], Array<string>)
* Tuples
* Enums
* Type assertions (as keyword)
* Type inference

**🧩 3. Functions**

* Function parameter and return types
* Optional and default parameters
* Rest parameters
* Anonymous and arrow functions
* Function overloading

**👨‍🏫 4. Object-Oriented Programming**

* Classes and objects
* Constructors
* public, private, protected, readonly modifiers
* Inheritance
* Abstract classes
* Interfaces with classes
* super and this keyword
* Static members

**🧱 5. Interfaces and Types**

* Defining and using interfaces
* Optional and readonly properties
* Index signatures
* Function types
* Extending interfaces
* Type aliases (type)
* Union and intersection types

**🧠 6. Advanced Types**

* Type narrowing
* Type guards (typeof, instanceof, custom guards)
* Discriminated unions
* Literal types
* Mapped types
* Conditional types
* Utility types (Partial, Required, Pick, Omit, Readonly, etc.)
* Template literal types

**🌐 7. Modules and Namespaces**

* ES6 module syntax (import / export)
* Default vs named exports
* Internal and external modules
* Namespaces (rarely used, but good to know)

**🏗️ 8. Generics**

* Introduction to generics
* Generic functions
* Generic classes
* Generic constraints (extends)
* Using keyof, in, and typeof with generics

**🛠️ 9. Tooling and Configuration**

* tsconfig.json in-depth
* TypeScript compiler options (strict, noImplicitAny, etc.)
* Source maps and debugging
* Working with third-party libraries
* Using DefinitelyTyped (@types)
* Linting with TSLint or ESLint

**⚙️ 10. Working with JavaScript & Node.js**

* Migrating JS to TS
* Mixing JS and TS
* Type declaration files (.d.ts)
* Node.js with TypeScript (using ts-node)
* Express.js with TypeScript

**🌐 11. TypeScript with Frameworks**

* [React](https://chatgpt.com?q=React) + TypeScript
* [Angular](https://chatgpt.com?q=Angular) + TypeScript (Angular is TypeScript-based)
* [Vue](https://chatgpt.com?q=Vue) + TypeScript
* Redux Toolkit with TypeScript
* REST API typing with Axios or Fetch

**🧪 12. Testing in TypeScript**

* Unit testing with [Jest](https://chatgpt.com?q=Jest)
* Mocking and typing
* Setting up tests with TypeScript

**💼 13. Real-World Projects & Practice**

* Building a simple ToDo App
* Building a CRUD app with API
* Type-safe forms
* Working with external APIs (with interfaces)

**🧪 Projects for Practice**

1. To-Do App (DOM Manipulation + Interfaces)
2. CRUD API with Express + TS
3. Weather App using Fetch API and Types
4. Blog app using React + TypeScript
5. Typing a real-world NPM package