Document Generated: 12/30/2025

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 2 Days

Introduction to TypeScript: Clean Code and Strong Skills for Web Developers (TT4154)

About This Course:

TypeScript helps you write more structured, reliable, and scalable code by adding strong typing and modern programming features to JavaScript. If you have some experience with web development or scripting and are looking to improve how you

organize, test, and maintain your code, this course is a practical next step. With expert guidance throughout, you will get the support you need to understand not just how TypeScript works, but why it makes a difference in real projects.

Over two days, you will build skills that make your everyday development work easier and more effective. You will learn how to write clear, type-safe functions, build and use custom types, and use interfaces to make your code more flexible and consistent across teams. The course will show you how to use classes and inheritance to reduce repetition, structure your code with modules, and improve error handling. You will also explore asynchronous programming, learning how to write and manage promises and use async and await to keep your code clean and readable. Along the way, you will get hands-on with tools like ts-jest for testing and InversifyJS for managing dependencies.

This course is designed for developers who want to sharpen their skills with modern tooling and best practices. It focuses on real-world use, giving you practical techniques you can take back to your projects right away. With about half the time spent on guided hands-on work, you will build confidence through practice and leave with a deeper understanding of how to write better software using TypeScript.

Course Objectives:

Geared for experienced web developers, this introductory-level course is focused on helping you get comfortable using TypeScript in the kind of projects you already work on. With expert guidance and plenty of hands-on time, you will build practical skills that make your code easier to understand, share, and keep working as it grows.

Working in a hands-on learning environment, guided by our expert instructor, you will learn how to:

- Use TypeScript type system to catch bugs early and make your code more predictable.
- Create your own types, interfaces, and classes to stay organized and reduce confusion in your projects.
- Write modern, cleaner functions with features like arrow functions and type inference.
- Handle async code with confidence using promises and async/await to keep things readable and smooth.
- Add structure with decorators and dependency injection so your code scales better as projects grow.
- Test your work with tools like ts-jest to make sure your TypeScript code is reliable and ready for real use.

Audience:

- This course is a great fit if you already know a bit of JavaScript or have worked on web or scripting projects and are ready to take the next step. It is designed for developers who want to write clearer, more manageable code using modern tools and techniques. Ideal attendees might be front-end developers, full-stack engineers, or anyone who wants to bring more structure to their coding.

Prerequisites:

This course is a great match for experienced web developers who have some hands-on experience with JavaScript or scripting and want to build more structured, reliable, and modern code using TypeScript. It is especially helpful for front-end developers, full-stack developers, and software engineers who work on web apps or team-based projects and are looking to improve code quality and maintainability. The course is approachable and guided by an expert, so you do not need to be advanced, but you should feel comfortable writing and understanding basic code.

Skills-based prerequisites:

- Able to write and read JavaScript code confidently

  Familiar with programming basics like variables, functions, and objects

  Have worked on a web project, script, or similar coding task

Take Before: In order to gain the most from this course, you should have incoming skills equivalent to those in the course listed below, or should have attended this as a prerequisite:

- Introduction to JavaScript / Modern JavaScript Essentials

Course Outline:

1.TypeScript Fundamentals

- Getting Started with TypeScript
- TypeScript and Functions
- TypeScript and Objects
- Basic Types
- Making Your Own Types

2.Declaration Files

- Declaration Files

- Third-Party Code Libraries

3.Functions

- Functions in TypeScript
- Function Expressions
- Arrow Functions
- Type Inference
- Functional Programming
- Organizing Functions into Objects and Classes
- Import, Export, and Require
- Unit Testing with ts-jest
- Error Handling

4.Classes and Objects

- What Are Classes and Objects?
- TypeScript Interfaces
- Generating HTML Code in Methods

5.Interfaces and Inheritance

- Interfaces
- TypeScript Inheritance

6.Advanced Types

- Type Aliases
- Type Literals
- Intersection Types
- Union Types
- Index Types

7.Decorators

- Reflection
- Importance of Decorators
- Decorators and Decorator Factories
- Class Decorators
- Method and Accessor Decorators
- Using Metadata in Decorators
- Property Decorators
- Parameter Decorators
- Application of Multiple Decorators on a Single Target

8.Dependency Injection in TypeScript

- The DI Design Pattern
- InversifyJS

9.Generics and Conditional Types

- Generics
- Conditional Types

10.Event Loop and Asynchronous Behavior

- The Multi-Threaded Approach
- Executing JavaScript
- Browsers and JavaScript
- Environment APIs
- Promises

11.Higher-Order Functions and Callbacks

- Higher-Order Functions
- Callbacks
- The Event Loop

12.Guide to Promises in TypeScript

- The Evolution of and Motivation for Promises
- Anatomy of a Promise
- Enhancing Promises with Types
- Libraries and Native Promises â€" Third-Party Libraries, Q, and Bluebird
- Promisify
- Asynchronous FileSystem
- Working with Databases
- Developing with REST
- Putting It All Together â€" Building a Promise App

13.Async/Await in TypeScript

- Evolution and Motivation
- async/await in TypeScript
- Syntax
- Exception Handling
- Top-Level await
- Promise Methods