

TypeScript & Angular

Course Title: Mastering Angular for Modern Web Development

Objective:

- To introduce students to the fundamental and advanced concepts of Typescript
- To provide hands-on experience in developing scalable and maintainable web applications using Angular.
- To equip students with the skills needed to integrate Angular with backend services, manage application state, and optimize performance.
- To enable students to become proficient in using Angular in real-world projects and prepare them for job opportunities in front-end development.

Introduction:

This course covers both TypeScript and Angular 17, two critical technologies for modern web application development. TypeScript adds type safety and advanced tooling support to JavaScript, enhancing developer productivity and code quality. Angular, a powerful front-end framework, enables the creation of robust, maintainable SPAs. The course combines theoretical concepts with hands-on labs to prepare students to confidently use TypeScript and Angular to build scalable applications.

Course Duration

- **Total Duration:** Approximately 10 weeks
- **Weekly Sessions:** 3 sessions per week, 2 hours per session

Course Outcomes:

By the end of this course, students will be able to:

1. Understand TypeScript fundamentals, including types, classes, interfaces, and decorators.
2. Build, structure, and scale applications effectively using TypeScript.
3. Develop SPAs using Angular 17, leveraging its latest features and TypeScript integration.
4. Utilize Angular services, components, routing, and modules to create organized and maintainable applications.
5. Deploy Angular applications with performance optimization and best practices.
6. Be ready to work on real-world, large-scale web application projects.

Why Should Students Learn Angular?

- **High Demand:** Angular is one of the most popular frameworks for front-end development, used by major companies worldwide.
- **Versatile Skillset:** Learning Angular alongside TypeScript equips students with a skill set that applies to enterprise and personal projects alike.
- **Scalability:** Angular is ideal for large applications and teams, with features designed for modular, maintainable codebases.
- **Performance and Efficiency:** Angular provides built-in tools and techniques to optimize applications, making them fast and responsive.
- **Community and Support:** Angular has a strong developer community and extensive resources for continued learning and problem-solving.

Syllabus Details :

A. TypeScript

Module 1: Introduction to TypeScript

- What is TypeScript?
- Benefits of TypeScript over JavaScript
- Setting up the TypeScript environment
- Compiling TypeScript code

Module 2: Basic Types and Type Annotations

- Primitive types and type annotations
- Type inference and explicit types
- Union and intersection types
- Literal types and type aliases

Module 3: Functions and Interfaces

- Function types and optional parameters
- Introduction to interfaces
- Using interfaces with functions
- Extending interfaces

Module 4: Classes and Object-Oriented Programming in TypeScript

- Defining classes, constructors, and properties
- Access modifiers (public, private, protected)
- Inheritance, polymorphism, and abstract classes
- Static properties and methods

Module 5: Advanced TypeScript Features

- Generics and generic constraints
- Enums and tuple types
- Type guards and type assertions
- Utility types and conditional types

Module 6: Decorators and Metadata

- Introduction to decorators
- Class and method decorators
- Practical use cases for decorators
- Metadata reflection and its applications

Module 7: Modules and Namespaces

- Working with ES6 modules in TypeScript
- Importing and exporting modules
- Namespaces and encapsulation
- Organizing code effectively

B. Angular 17 Modules

• Module 1: Introduction to Angular 17 and Setup

- What is Angular and how it differs from other frameworks
- Setting up Angular CLI and project structure
- Overview of Angular's architecture and core modules

• Module 2: Components and Data Binding

- Creating and organizing components
- Component templates, styles, and encapsulation
- Data binding (property, event, two-way binding)
- Component communication (Input and Output)

- **Module 3: Directives and Pipes**
 - Structural and attribute directives
 - Creating custom directives
 - Angular pipes for data transformation
 - Building custom pipes
- **Module 4: Services and Dependency Injection**
 - Creating and injecting services
 - Angular's dependency injection system
 - Managing state with services
 - Using services for HTTP requests and API integration
- **Module 5: Angular Routing and Navigation**
 - Setting up routing and defining routes
 - Route parameters and query parameters
 - Lazy loading and route guards
 - Nested routes and advanced routing features
- **Module 6: Reactive Forms and Template-driven Forms**
 - Template-driven vs reactive forms
 - Form validation and error handling
 - Dynamic form controls and custom validators
 - Best practices for managing forms
- **Module 7: State Management and RxJS**
 - Introduction to RxJS and observables
 - Managing state with BehaviorSubject and Observables
 - Using NgRx for complex state management
 - Advanced reactive programming techniques

- **Module 8: Angular Performance Optimization and Best Practices**

- Lazy loading, preloading strategies, and module splitting
- Change detection strategies
- Caching, optimization, and debugging
- Testing Angular applications (unit and integration testing)

- **Module 9: Deploying and Maintaining Angular Applications**

- Building and deploying applications
- Working with Angular environments
- Deployment strategies and tools
- Application maintenance and upgrade considerations

Career Opportunities After Learning TypeScript and Angular

- **Front-end Developer:** Design and develop dynamic front-end web applications, using TypeScript and Angular to create user-friendly interfaces.
- **Full Stack Developer:** Combine TypeScript and Angular for the front end and integrate with back-end technologies, enhancing versatility.
- **UI/UX Engineer:** Use Angular's component-based architecture to build responsive and accessible interfaces.
- **Web Application Developer:** Create high-performance single-page applications (SPAs) for companies in sectors like tech, finance, healthcare, etc.
- **Angular Developer:** Specialized role focusing on Angular-based application development, essential for large-scale enterprise projects.
- **Application Architect:** Use TypeScript and Angular to design scalable and maintainable architectures for complex applications.
- **Technical Lead/Manager:** Guide teams in building Angular applications and ensuring code quality, performance, and scalability.