SE001TS –*Introduction to TypeScript*

Duration: 3 Days; Instructor-led

WHAT YOU WILL LEARN

TypeScript is a superset developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for the development of large applications and transcompiles to JavaScript.

TypeScript may be used to develop JavaScript applications for both client-side (as with Angular or React) and server-side execution (as with Node.js or Deno).

Many JavaScript based projects getting more popular and complex, Therefore, TypeScript start having it place in large JavaScript-based project development.

This course is specially designed to focus on the basic features and fundamentals of TypeScript. Upon completion of this course, learners will be equipped with the understanding of fundamental concepts and features of TypeScript. With this preparation, learners will be able to continue exploring more advance TypeScript features and demonstrate more effective coding ability.

AUDIENCE

This course is specially designed for programmers who are new to TypeScript.

PREREQUISITES

- Familiar with Windows Operating Systems.
- Basic programming skill and knowledge are not mandatory but will be added advantage.

METHODOLOGY

This program will be conducted through Instructor-led (classroom)

COURSE OBJECTIVES

After completing this course, students will be able to:

- differentiate TypeScript and JavaScript
- create TypeScript projects using IDE
- understand the basic programming construct with TypeScript
- apply different operators
- write code using different programming flow constructs
- appreciate the importance of OOP in TypeScript program development
- support Information Hiding in TypeScript
- deal with different parameter declarations
- use Array to keep heterogenous data types
- resolve ambiguity with namespace
- decide when to use internal and external modules
- define Ambients

COURSE OUTLINE

Module 1 – Overview

- What is TypeScript?
- Features of TypeScript
- Why Use TypeScript?
- Components of TypeScript

Module 2 – Environment Setup

- Local Environment Setup
- Installing Node.js
- IDE Support
- Visual Studio Code

Module 3 –Basic Syntax

- Your First TypeScript Code
- Compile and Execute a TypeScript Program
- Compiler Flags
- Identifiers in TypeScript
- TypeScript Keywords
- Comments in TypeScript
- TypeScript and Object Orientation

Module 4 - Types

- The Any type
- Built-in types
- User-defined Types

Module 5 – Variables

- Variable Declaration in TypeScript
- Type Assertion in TypeScript
- Inferred Typing in TypeScript
- TypeScript Variable Scope

Module 6 – Operators

- What is an Operator?
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Miscellaneous Operators
- Type Operators

Module 7 – Decision Making

- If-Then
- If-Then-Else
- Else-If
- Switch statement

Module 8 - Loops

- Definite Loop
- Indefinite Loop
- The break Statement
- The continue Statement
- The Infinite Loop

Module 9 - Functions

- Optional Parameters
- Rest Parameters
- Default Parameters

- Anonymous Function
- The Function Constructor
- Recursion and TypeScript Functions
- Lambda Functions
- Syntactic Variations
- Function Overloads

Module 10 – Numbers
- Syntax
- Number Methods

Module 11 – Strings
- Syntax
- String Methods

Module 12 – Arrays
- Features of an Array
- Declaring and Initializing Arrays
- Accessing Array Elements
- Array Object
- Array Methods
- Array Destructuring
- Array Traversal using for…in loop
- Arrays in TypeScript

Module 13 – Tuples
- Syntax
- Accessing values in Tuples
- Tuple Operations
- Updating Tuples
- Destructuring a Tuple

Module 14 – Union
- Syntax
- Union Type and Arrays

Module 15 – Interfaces
- Declaring Interfaces
- Union Type and Interface
- Interfaces and Arrays
- Interfaces and Inheritance

Module 16 – Classes
- Creating classes
- Creating Instance objects
- Accessing Attributes and Functions
- Class Inheritance
- Class inheritance and Method Overriding
- The static Keyword
- The instanceof operator
- Data Hiding
- Classes and Interfaces

Module 17 – Objects
- Syntax
- TypeScript Type Template
- Duck-typing

Module 18 – Namespaces
- Defining a Namespace
- Nested Namespaces

Module 19 – Modules
- Internal Module
- External Module

Module 20 – Ambients
- Purpose
- Defining Ambients