

**Course Name** : PRACTICAL TYPESCRIPT  
**Duration** : 3 Days (Physical Classroom / Virtual Live Instructor)  
**Skill Level** : Beginner

#### **COURSE DESCRIPTION:**

The TypeScript course is a program designed to provide individuals with the skills and knowledge needed to utilize the TypeScript language in their web development projects. Through practical training and hands-on exercises, participants will learn about the key features of TypeScript, including type annotations, classes, modules, and decorators. They will also learn how to use TypeScript in combination with popular JavaScript frameworks such as Angular, React, and Vue. Upon completion of the course, participants will be able to develop robust and scalable applications using TypeScript.

#### **WHAT WILL YOU LEARN?**

In this Practical TypeScript course, our HRDF-certified instructor will guide you in harnessing the full potential of the latest TypeScript language in your JavaScript applications. The program will begin with a review of essential JavaScript concepts, followed by an in-depth examination of TypeScript's distinct features including data types, classes, generics, modules, and decorators.

Upon completion of the course, candidates will possess a strong foundation to commence the development of applications utilizing TypeScript independently or in conjunction with popular JavaScript frameworks such as Angular, Vue3, and React.

#### **PREREQUISITE:**

Beginners. JavaScript helps but is not mandatory.

#### **METHODOLOGY:**

This program will be conducted with interactive lectures, PowerPoint presentations, discussions, and practical exercises. This course can be conducted as instructor-led (ILT) or virtual instructor-led training (VILT).

#### **JOB SCOPE:**

Upon completion of this course, candidates may pursue the following career paths:

- Front-end Web Developer
- Full Stack Developer
- JavaScript Developer
- Web Application Developer
- Enterprise Application Developer
- Technical Lead/Manager in a web development team
- Angular Developer
- React Developer
- Vue Developer
- Mobile App Developer (using frameworks such as Ionic)
- Cloud Developer (utilizing TypeScript for serverless applications)

## DAY 1

### **MODULE 1: GETTING STARTED WITH TYPESCRIPT**

- Welcome
- Introduction to TypeScript
- Installation
- Executing TypeScript Code
- Catching Errors with TypeScript

### **MODULE 2: INTRO TO TYPE STREAM**

- Introduction
- Types
- More on Types
- Examples of Types
- Where Do We Use Types?

### **MODULE 3: TYPE ANNOTATIONS IN ACTION**

- Type Annotations and Inference
- Annotations with Variables
- Object Literal Annotations
- Annotations Around Functions
- Understanding Inference
- The 'Any' Type
- Fixing the 'Any' Type
- Delayed Initialization
- When Inference Doesn't Work?

## DAY 2

### **MODULE 4: ANNOTATIONS FUNCTIONS AND OBJECTS**

- Annotations Around Functions
- Inference Around Functions
- Annotations for Anonymous Functions
- Void and Never
- Destructuring with Annotations
- Annotations Around Objects

### **MODULE 5: MASTERING TYPED ARRAYS**

- Why Typed Arrays?
- Multiple Types in Arrays
- When to Use Typed Arrays

## MODULE 6: TUPLES IN TYPESCRIPT

- Introduction to Tuples
- Tuples in Action
- Why Tuples?

## DAY 3

## MODULE 7: INTERFACES IN TYPESCRIPT

- Introduction to Interfaces
- Long Type Annotations
- Fixing Long Annotations with Interfaces
- Syntax Around Interfaces
- Functions in Interfaces
- Code Reuse with Interfaces
- General Plan with Interfaces

## MODULE 8: BUILD FUNCTIONALITY WITH CLASSES

- Classes
- Basic Inheritance
- Instance Method Modifiers
- Fields in Classes
- Fields with Inheritance
- Where to Use Classes

## MODULE 9: DESIGN PATTERN & BUILD AN WEB APP

- App Overview
- Bundling with Parcel
- Project Structure
- Generating Random Data
- Type Definition Files
- Using Type Definition Files
- Export Statements in Typescript
- Defining a Company
- Note on Generating an API Key
- Adding Google Maps Support
- Google Maps Integration
- Exploring Type Definition Files
- Hiding Functionality
- Why Use Private Modifiers? Here's Why
- Adding Markers
- Duplicate Code
- One Possible Solution
- Restricting Access with Interfaces
- Implicit Type Checks

- Showing Popup Windows
- Updating Interface Definitions
- Optional Implements Clauses
- App Wrap up

## CONCLUSION

- QA
- Useful References and Books
- Feedback