



Essential TypeScript 2.0 with Visual Studio Code

Duration: 5 Days **Course Code: GK4544**

Overview:

TypeScript is a superset of JavaScript that compiles to plain JavaScript that can run in any browser and on any platform. It supports the latest JavaScript features, including those from ECMAScript 2015, as well as proposed features from future JavaScript versions, such as decorators and `async / await`. But what makes TypeScript ideal for large web applications is the addition of optional type annotations, which allow developer tools to provide intellisense and syntax checking, so that errors are caught at compile-time instead of at run-time. This course provides in-depth coverage of TypeScript, including language syntax, object-oriented and asynchronous programming, and how to organize your code using namespaces and modules. You will also learn to use Visual Studio Code, Microsoft's lightweight cross-platform code editor, to build RESTful services in TypeScript with Node.js, as well as responsive client applications with Angular 2, which is itself written in TypeScript.

Note: You may use your own laptop with Windows, OS X or Linux.

Target Audience:

Developers who want to build client or server JavaScript applications using TypeScript with Visual Studio Code for improved productivity with intellisense, debugging and code refactoring

Objectives:

- Compile TypeScript to JavaScript with support for intellisense and debugging
 - TypeScript syntax for variable declarations, non-nullable types, operators and flow control statements
 - Tips and tricks for using Visual Studio Code to build TypeScript applications
 - Configure Visual Studio Code for task automation with Gulp
 - Lint and compile TypeScript code with extensions for Visual Studio Code
 - Write unit tests in Jasmine for behavior and test driven development
 - Use Karma for Continuous Integration to run unit tests as part of a build process
 - TypeScript type system, union and intersection types, type guards and string literal types
 - Arrow functions, function overloading, optional, default and rest parameters
 - Asynchronous programming with promises, generators and `async / await`
 - Object-oriented programming with interfaces, classes, inheritance and class expressions
 - Algorithm reuse with generic interfaces, classes and functions with generic constraints
 - Namespaces and modules with support for various module loaders, including ECMAScript 2015 modules
 - Build RESTful services in TypeScript with Node.js and Express
 - Build Single Page Applications (SPA's) in TypeScript using Angular 2
-

Prerequisites:

- Experience with JavaScript or an object-oriented programming language such as Java or C#
-

Content:

Course Outline:

Module 1: Introduction to TypeScript

- History and Importance of JavaScript
- JavaScript Strengths, Weaknesses
- Limitations of JavaScript
- Advantages of TypeScript
- TypeScript Design Goals
- Installing TypeScript
- Getting Started

Module 2: TypeScript Language Basics

- Declaring Variables
- Basic Types
- Type Annotations
- Non-Nullable Types
- Basic Data Structures
- Operators
- Flow Control Statements

Module 3: Using Visual Studio Code for TypeScript Development

- Tooling Options
- VS Code Features
- Limitations
- Project System
- Navigation Basics
- Productivity
- JavaScript Intellisense
- Configuring TypeScript Projects
- Compiling TypeScript
- Debugging

Module 4: Task Automation, Unit Testing, Continuous Integration

- Need for Task Automation
- Task Runners
- Introduction to Gulp
- Using Gulp
- Configuring Build Tasks
- Debugging Gulp Tasks
- Testing Approaches
- Testing Frameworks
- BDD with Jasmine
- Running Tests in a Browser
- Writing Tests in TypeScript
- Test Runners
- Continuous Integration

Module 5: TypeScript Type System

- Type Compatibility
- Type Inference
- Access Modifiers
- Union Types
- Type Guards
- Intersection Types
- Type Aliases
- String Literal Types

Module 8: Object-oriented Programming

- Introduction to Object-Oriented Programming
- SOLID Principles
- Interfaces
- Classes
- Inheritance
- Mixins
- Abstract Classes
- Class Expressions

Module 9: Generics

- Algorithm Reuse
- Generic constructs
- Type Parameters
- Constraints
- Generic Functions
- Generic Interfaces and Classes

Module 10: Namespaces and Modules

- Name Collisions
- Namespaces
- Defining Namespaces
- Limitations of HTML Script Tags
- Module Loaders
- ECMAScript 2015 Module Support
- Module Organization
- Exporting from Modules
- Importing from Modules
- Module Guidelines

Module 11: Practical TypeScript with Express and Angular

- SPA Architecture
- Introduction to REST
- Express Framework
- Routing Basics
- MV* Architecture
- Web Client Frameworks
- Angular 2 Concepts
- Angular Components
- Angular Templates
- Angular Metadata
- TypeScript Decorators
- Services and Dependency Injection
- HTTP Requests

Labs:

Lab 1: Introduction to TypeScript

- Install TypeScript using NPM
- Convert JavaScript into TypeScript
- Compile TypeScript into JavaScript.

Lab 2: TypeScript Language Basics

- Use var, let and const to declare variables

Lab 4: Task Automation, Unit Testing, Continuous Integration

- Install Gulp using NPM
- Add gulpfile.js, import gulp plugins, add type declarations
- Set up Gulp debugging
- Configure VS Code to use Gulp for compiling TypeScript
- Write unit tests using Jasmine, serve tests using browser-sync, run tests using Karma
- Configure continuous integration with TravisCI

Lab 5: TypeScript Type System

- Use structural typing
- Type inference
- Access modifiers
- Union types
- Type guards
- Intersection types

Lab 6: Functional Programming

- Use named functions, arrow functions and function types
- Optional and default parameters
- Rest parameters and function overloading

Lab 7: Asynchronous Programming

- Callbacks
- Promises
- Generators
- async / await

Lab 8: Object-oriented Programming

- Use interfaces and classes with inheritance and mixins
- Define abstract classes and class expressions

Lab 9: Generics

- Create generic functions, interfaces and classes
- Apply generic constraints
- Use generic entities

Lab 10: Namespaces and Modules

- Use namespaces to organize code for small client applications
- Use modules to organize code for large applications
- Use ES 2015 module loader at design-time and runtime
- Export and import from modules

Lab 11: Practical TypeScript with Express and Angular

- Polymorphic "this" Types

Module 6: Functional Programming

- Functional Programming
- Named Functions
- Arrow Functions
- Function Types
- Optional and Default Parameters
- Rest Parameters
- Function Overloading

Module 7: Asynchronous Programming

- Importance of Async
- Callbacks
- Callback Hell
- Promises
- Generators
- Async and Await

- Use type annotations to declare basic types explicitly
- Create basic data structures and use operators with flow control statements

Lab 3: Using Visual Studio Code for TypeScript Development

- Launch Visual Studio Code
- Scaffold a project with Yeoman
- Import type declarations for intellisense with JavaScript libraries
- Configure a TypeScript project, specify compilation options
- Add a build task, auto compile when files change
- Configure debugging, set breakpoints, add watches for variables and expressions

- Use Express with Node.js to build a RESTful web service
- Build an Angular 2 web application with components, templates and metadata
- Render views with data returned by HTTP requests to the Express web service

Further Information:

For More information, or to book your course, please call us on Head Office 01189 123456 / Northern Office 0113 242 5931

info@globalknowledge.co.uk

www.globalknowledge.co.uk

Global Knowledge, Mulberry Business Park, Fishponds Road, Wokingham Berkshire RG41 2GY UK