Introduction to TypeScript

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **2 Gün**

https://bilginc.com/tr/egitim/introduction-to-typescript-5604-egitimi/

Overview

JavaScript is a powerful language that is ubiquitous in the Web. However, its dynamic and interpreted nature can lead to problems at scale, problems that are mitigated in other languages by compilation and static typing. TypeScript brings these benefits to the JavaScript universe.

TypeScript is a type safe superset of JavaScript that allows us to write clean, simple code that runs anywhere JavaScript does. Unlike other languages that can transpile to JavaScript, the interop between JavaScript and TypeScript is a primary engineering concern. As a superset, all JavaScript skills and legacy code are readily reusable. Type safety brings many benefits, including catching bugs at compile time instead of run time, advanced static analysis tooling, powerful refactorings, better IDE experience etc.

INSTIL

Prerequisites

Delegates must be proficient JavaScript developers. However, the delivery time can be extended if have no prior knowledge of JavaScript.

What You Will Learn

- Understand the importance of TypeScript and where it fits in the web ecosystem
- Build a firm understanding of all core TypeScript features
- Be comfortable applying both Object Oriented and Functional paradigms
- Become comfortable with Algebraic Data Types and other advanced concepts

Outline

Introduction and Setup

- Why do we need TypeScript?
- What is Transpilation
- The difference between the TypeScript compiler and Babel
- Setting up a build environment with Webpack
- Support for TS in the IDE
- Configuring TypeScript via *tsconfig.json*
- Decoding the transpiled output

TypeScript Fundamental

- The basic types supported by TS
- The power of type inference
- Comparing *any* and *unknown*
- The difference between Casting and Type Assertions
- TypeScript as a superset of Modern JavaScript
- Enhancements to function declarations
- Using function types and higher order functions
- How function overloading works in TypeScript
- Numeric, String and *const* enumerations

Object Oriented TypeScript

- The TypeScript class model in depth
- Access modifiers and support for JavaScript private
- Enforcing contracts via interfaces
- Readonly properties and Accessors
- Support for generic types and constraints

Advanced TypeScript

- Advanced features of TS interfaces
- The difference between object literals and type literals
- Using string and number literals as Types
- Type Aliases, Unions and Intersections
- Smart Casting and Type Guards
- Defining mapped types and the *infer* keyword
- Utility types and type programming
- *ts-toolbelt* and *utility-types*
- Using class Decorators
- Understanding Symbols
- Generator functions
- Writing asynchronous TypeScript

Testing TypeScript

- Mock objects in a strongly typed world
- Leveraging the Builder pattern
- Testing asynchronous TypeScript