

TypeScript 튜토리얼

정보통신공학부 2018036044 이승준

목차

1. TypeScript 란 무엇인가?
2. TypeScript 사용 방법

TypeScript가 무엇인가?

TypeScript는 Microsoft에서 개발한 오픈 소스 프로그래밍 언어로, JavaScript에 정적 타입을 추가한 슈퍼셋(Superset)입니다. TypeScript는 JavaScript 코드를 작성할 수 있는 모든 환경에서 사용할 수 있으며, 정적 타입 시스템을 통해 코드의 가독성과 유지보수성을 높이는 데 중점을 두고 있습니다.

TypeScript의 주요 특징과 개념은 다음과 같습니다:

1. 정적 타입 시스템:

- TypeScript는 변수, 함수, 객체 등의 타입을 명시적으로 지정할 수 있습니다. 이로써 컴파일러가 코드를 분석하고 런타임 이전에 타입 오류를 감지할 수 있습니다. 이는 코드의 신뢰성을 높이고 디버깅을 용이하게 만듭니다.

2. 인터페이스와 타입 별칭:

- TypeScript에서는 인터페이스와 타입 별칭을 사용하여 사용자 정의 타입을 선언할 수 있습니다. 이는 코드의 가독성을 높이고 코드베이스를 추상화할 때 유용합니다.

3. 클래스와 모듈:

- TypeScript는 클래스와 모듈을 지원하여 객체지향 프로그래밍 및 모듈화된 코드 구조를 작성할 수 있도록 합니다.

4. 제네릭(Generics):

- 제네릭을 사용하여 코드의 재사용성을 높일 수 있습니다. 함수나 클래스를 작성할 때 타입이나 데이터 구조를 파라미터화하여 다양한 타입에서 사용할 수 있게 합니다.

5. 타입 가드(Type Guards):

- TypeScript에서는 타입 가드를 사용하여 런타임에서 특정 타입을 검사하고 타입을 좁힐 수 있습니다.

TypeScript는 JavaScript의 상위 집합으로 사용되며, TypeScript 코드는 JavaScript로 컴파일 될 수 있습니다. 특히 대규모 프로젝트에서 타입스크립트를 사용하면 코드의 유지보수성을 높일 수 있고, 팀 프로젝트에서 협업이 더 쉬워집니다.

TypeScript 사용방법

1. TypeScript 설치

먼저, 프로젝트 폴더에서 다음 명령어를 사용하여 TypeScript를 설치합니다.

```
npm install --save-dev typescript 또는 yarn add --dev typescript
```

2. tsconfig.json 파일 생성

프로젝트 루트에 tsconfig.json 파일을 생성하여 TypeScript 컴파일러의 설정을 정의합니다. 이 파일은 TypeScript 컴파일 옵션 및 프로젝트 구성을 설정하는 데 사용됩니다.

```
// tsconfig.json
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "moduleResolution": "node",
    "jsx": "react"
  }
}
```

target: 컴파일된 JavaScript 코드의 ECMAScript 버전을 지정합니다.

module: 모듈 시스템을 지정합니다. 대부분의 경우 "commonjs"를 사용합니다.

strict: 엄격한 타입 체크를 활성화합니다.

esModuleInterop: CommonJS 모듈 시스템에서 esModuleInterop 옵션을 사용합니다.

skipLibCheck: 라이브러리 검사를 건너뛰도록 설정합니다.

forceConsistentCasingInFileNames: 파일 이름의 일관된 대소문자 사용을 강제합니다.

moduleResolution: 모듈 해결 방식을 지정합니다.

jsx: JSX 코드를 사용할 때 React에 맞는 설정을 지정합니다.

3. TypeScript 파일 작성

이제 TypeScript 파일 (확장자 .ts 또는 .tsx)을 작성할 수 있습니다.

```
// example.ts
function greet(person: string): string {
  return `Hello, ${person}!`;
}

const result = greet('TypeScript');
console.log(result);
```

4. TypeScript 파일 컴파일

프로젝트 루트에서 다음 명령어를 사용하여 TypeScript 파일을 JavaScript로 컴파일합니다.

`npx tsc` 또는 `yarn tsc`

5. 실행

컴파일된 JavaScript 파일을 실행합니다.

`node example.js`

이제 TypeScript를 사용하여 JavaScript 코드를 작성하고, TypeScript 컴파일러를 사용하여 JavaScript로 변환할 수 있습니다. 만약 React 프로젝트를 진행한다면, 해당 프로젝트에 맞게 tsconfig.json 파일을 구성하고 JSX 설정을 추가하면 됩니다.