

Migrating from JavaScript to TypeScript

TypeScript. JavaScript that scales.

Seungjae Lee @huisseoul

Huiseoul?

- Front-end
 - React Native, Mobx ...
- Back-end
 - Node.js, GraphQL ...
- Written by TypeScript and one @types/huiseoul

History

- 2012년 10월 MicroSoft에서 발표
 - 개발툴은 윈도우만 지원
- 2014년 4월 Build 행사에서 v1.0 발표
 - VS2013에 built-in
- 2014년 7월 새로운 컴파일러 소개
 - 5배 빨라짐, Github으로 소스 이전
- 2016년 9월 v2.0 발표
 - @types/* 패키지 풀기 시작
 - --strictNullChecks 추가
- 현재 v2.3.2

Specs

- Open-source: <https://github.com/Microsoft/TypeScript>
- Superset of JavaScript
 - JavaScript 문법 그대로 사용 가능
- Static Typing
 - Checked at compile time
 - Type annotation
- OOP
- EcmaScript features
 - supports v3 ~ future proposals
- ... and much more

Benefit

- 정적 타이핑이 갖는 장점들
 - 안정적: 대규모 프로젝트에서도 OK
 - 명시적: 코드 잘 읽힘 -> 관리 용이
- Learning curve
 - JavaScript 다룰 줄 알면 금방 익숙해짐
- JavaScript 패키지 그대로 사용 가능
- 쉽다!

Install & run

```
$ npm install -g typescript
$ echo 'console.log("Hello, World!");' > saystone.ts
$ tsc saystone.ts
$ node saystone.js
Hello, World!
```

Compiler config

- `your-project/tsconfig.json`

```
# automatically points ./your-project/tsconfig.json
your-project$ tsc
```

- <https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>
- <http://json.schemastore.org/tsconfig>

Example: Pure TypeScript

```
class Student {
  fullName: string;
  constructor(public firstName, public middleInitial, public lastName) {
    this.fullName = firstName + " " + middleInitial + " " + lastName;
  }
}

interface Person {
  firstName: string;
  lastName: string;
}

function greeter(person : Person) {
  return "Hello, " + person.firstName + " " + person.lastName;
}

var user = new Student("Jane", "M.", "User");

console.log(greeter(user));
```

Output: Converted JavaScript

```
var Student = (function () {
    function Student(firstName, middleInitial, lastName) {
        this.firstName = firstName;
        this.middleInitial = middleInitial;
        this.lastName = lastName;
        this.fullName = firstName + " " + middleInitial +
    }
    return Student;
})();
function greeter(person) {
    return "Hello, " + person.firstName + " " + person.la
}
var user = new Student("Jane", "M.", "User");
console.log(greeter(user));
```


JavaScript to TypeScript

1. tic-tac-toe: React tutorial

- JavaScript + React

2. tic-tac-toe: 지난주 GeekTalk

- JavaScript + React + Redux

3. tic-tac-toe: 오늘 라이브 코딩

- TypeScript + React + Redux

We loves github

1. fork
2. clone
3. coding
4. commit
5. ...repeat
6. coding
7. commit
8. push
9. Pull request to origin repository

Step

1. 기존 프로젝트 실행해보기
 - tic-tac-toe 로직 슬쩍 훑어보기
2. TypeScript 설치 & 환경 설정
3. VSCode 빌드 설정
 - 일일이 파일별로 컴파일하지 않아도 되어요
 - `./.vscode/tasks.json`
4. 기존 `./src` 디렉토리를 `./ts` 로 변경
5. 모든 JavaScript 파일 확장자를 `ts` 로 변경
 - 컴파일 에러 뽀뽀 -> 파일 하나씩 열면서 수정
6. `*.css` 파일을 `./public` 으로 옮기고 `./public/index.html` 에 CSS 경로 지정
7. Happy coding

Before start

- `tic-tac-toe` 는 `react-scripts` 패키지를 사용합니다.
 - `create-react-app` 의 서브패키지
- `react-scripts` 패키지 안엔 Webpack, Babel, ESLint 등 모듈과 설정이 빌트인돼있어요.
 - bundling, pre-compile, local server, hot-reload 등
- 위 패키지가 JavaScript에만 맞춰져 있고 빌트인 돼있기 때문에 세부 설정까지 TypeScript에 맞게 바꿀순 없지만 컴파일하면 뽑히는 `*.js` 파일들을 위 패키지가 읽을수 있도록 덮어쓰는 방식으로 진행합니다.
- 추가로 게임 자체 로직이나 컨벤션은 신경쓰지 않는 쪽으로!
- 최대한 TypeScript 변환에 집중!

ETC

- TypeScript React Starter:
<https://github.com/Microsoft/TypeScript-React-Starter#typescript-react-starter>
- TSLint: <https://palantir.github.io/tslint/>
- @type/* packages:
<https://www.typescriptlang.org/docs/handbook/declaration-files/introduction.html>

References

- <https://medium.com/@gaperton/typescript-static-or-dynamic-64bceb50b93e>
- <https://hyunseob.github.io/2016/09/25/typescript-introduction/>
- <http://2ality.com/2014/10/typed-javascript.html>
- <https://github.com/Microsoft/TypeScript>
- <https://en.wikipedia.org/wiki/TypeScript>

감사합니다.

fin.