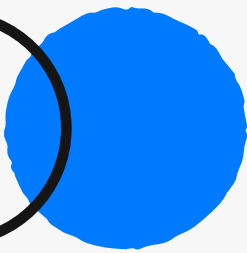




## Effective TypeScript

Item 51



# 의존성 분리를 위해 미리 타입 사용하기



@Bori-github

# 목차

01

라이브러리 만들기

02

구조적 타이핑 사용하기

03

마무리

O'REILLY®

## Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

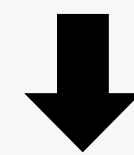
01

## 라이브러리 만들기

### ✓ CSV 파일을 파싱하는 라이브러리

```
1 function parseCSV(contents: string | Buffer): {[column: string]: string}[] {  
2     // ...  
3 }
```

- 매개변수로 CSV 파일의 내용(content), NodeJS 사용자를 위해 Beffer 타입을 허용
- Beffer 타입의 정의는 @type/node를 통해 얻어 **타입선언이 @type/node에 의존**



@type/node를 devDependencies에 포함

다음과 같은 사용자들을 고려

- 타입스크립트 비사용자 : 자바스크립트 개발자의 경우 @type과 무관
- NodeJS 비사용자 : Buffer는 NodeJS 개발자만 필요



@type/node를 devDependencies에 포함할 경우,  
두 그룹의 사용자들은 각자 사용하지 않는 모듈이 포함됨

⇒ @type/node는 NodeJS와 타입스크립트를 **동시에 사용하는 개발자만** 관련

## 구조적 타이핑 사용하기

✓ 구조적 타이핑을 적용하여 필요한 메서드와 속성만 별도로 작성

```
1 interface CsvBuffer {  
2     // Buffer 타입을 사용하지 않고, toString 메서드를 직접 구현  
3     toString(encoding: string): string;  
4 }  
5  
6 function parseCSV(contents: string | CsvBuffer): {[column: string]: string}[] {  
7     // ...  
8 }
```

- CsvBuffer 인터페이스를 사용하여 실제로 **필요한 부분만을 명시**
- **Beffer** 와 **호환**이 되어, NodeJS 프로젝트에서 parseCSV를 호출 가능

## 미러링(mirroring)

- **필요한 선언부만 추출**하여 작성 중인 라이브러리에 넣는 것
- 다른 라이브러리의 타입에만 의존하는 경우, 미러링을 적용 가능

다른 라이브러리의 타입 선언의 대부분을 추출해야한다면?

⇒ 명시적으로 **@type 의존성을 추가**

- 필수가 아닌 의존성을 분리할 때 **구조적 타이핑**을 사용할 수 있습니다.
- 공개 라이브러리를 사용하는 여러 그룹의 사용자가 각자 사용하지 않는 모듈의 의존성을 가지지 않게 해야합니다.



# Thanks!

**Do you have any questions?**

qhflrnfl4324@gamil.com

<https://github.com/Bori-github>

