

Item 20.

다른 타입에는 다른 변수 사용하기

---

**Effective TypeScript**

# 목 차

1. 변수 재사용
2. 타입 좁히기
3. 다른 변수 사용하기
4. 결론

O'REILLY®

# Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

# 1. 변수 재사용

자바스크립트에서 한 변수를 다른 목적을 가지는 다른 타입으로 **재사용** 가능



```
1 function fetchProduct(id: string) {}  
2 function fetchProductBySerialNumber(id: number) {}  
3  
4 let id = "12-34-56"; // string  
5 fetchProduct(id);  
6  
7 id = 123456; // number  
8 fetchProductBySerialNumber(id);
```

# 1. 변수 재사용

## 타입스크립트에서는?

⇒ id가 “12-34-56”이라는 값을 통해 **string**으로 추론되어 오류 발생

```
1 function fetchProduct(id: string) {}
2 function fetchProductBySerialNumber(id: number) {}
3
4 let id = "12-34-56"; // string
5 fetchProduct(id);
6
7 id = 123456; // number
8 fetchProductBySerialNumber(id);
```

'number' 형식은 'string' 형식에 할당할 수 없습니다. ts(2322)

```
let id: string
```

이 호출과 일치하는 오버로드가 없습니다.

오버로드 1/2('(id: number): void')에서 다음 오류가 발생했습니다.

'string' 형식의 인수는 'number' 형식의 매개 변수에 할당될 수 없습니다.

오버로드 2/2('(id: number): void')에서 다음 오류가 발생했습니다.

'string' 형식의 인수는 'number' 형식의 매개 변수에 할당될 수 없습니

다. ts(2769)

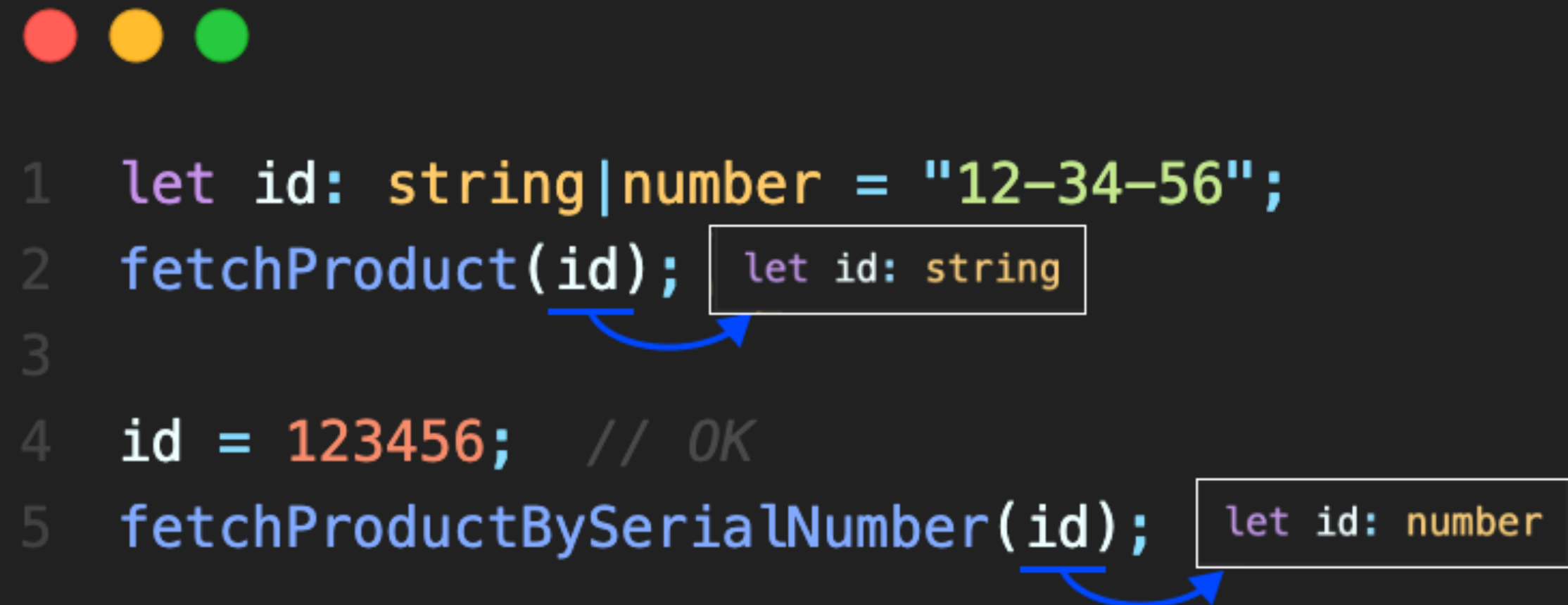
```
let id: string
```

**“변수의 값은 바뀔 수 있지만 그 타입은 보통 바뀌지 않는다”**

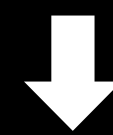
## 2. 타입 좁히기

### 타입을 변경하는 방법

범위를 좁혀 타입 변경 ⇒ 유니온 타입 활용하기



```
1 let id: string|number = "12-34-56";
2 fetchProduct(id);
3
4 id = 123456; // OK
5 fetchProductBySerialNumber(id);
```



id를 사용할 때마다 값이 어떤 타입인지 확인이 필요하다는 문제점 발생



### 3. 다른 변수 사용하기

무분별한 변수의 재사용은 타입 체커와 사람 모두에게 혼란

⇒ 다른 타입에는 **별도의 변수를 사용**하는 것이 바람직



```
1  const id = "12-34-56";  
2  fetchProduct(id);  
3  
4  const serial = 123456; // OK  
5  fetchProductBySerialNumber(serial); // OK
```

- 서로 관련이 없는 두 개의 값 분리
- 구체적인 변수명
- 타입 추론 향상 및 불필요한 타입 구문 제거
- 간결한 타입 정의
- let 대신 const 키워드를 사용한 변수 선언

### 3. 다른 변수 사용하기

주의할 점! 재사용 되는 변수와 **가려지는 변수**를 혼동하지 않기

가려지는 변수(Variable shadowing)

- 하나의 범위(scope) 내에서 동일한 이름의 변수가 중복해서 선언되는 현상
- 외부 범위에서 이미 선언된 변수와 동일한 이름의 변수가 내부 범위에서 선언되면, 내부 범위에서의 변수가 외부 범위에서의 변수를 “가리게” 되는 것을 의미



```
1  const id = "12-34-56";  
2  fetchProduct(id);  
3  
4  {  
5    const id = 123456; // OK  
6    fetchProductBySerialNumber(id); // OK  
7  }
```

⇒ 가려지는 변수로 인해 타입의 오류 발생하지 않음

## 4. 결론

변수의 값은 바뀔 수 있지만 타입은 일반적으로 바뀌지 않습니다.

타입이 다른 값을 다루게 될 경우, 변수의 타입을 변경하는 방법을 통해 변수를 **재사용** 할 수 있으나 다른 타입에는 다른 변수를 선언하여 사용하는 것이 바람직합니다.



감사합니다 :) )

Effective TypeScript

@Bori-github(이보리)