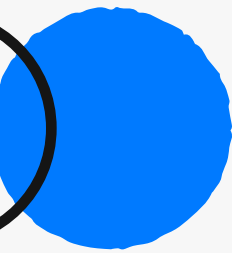


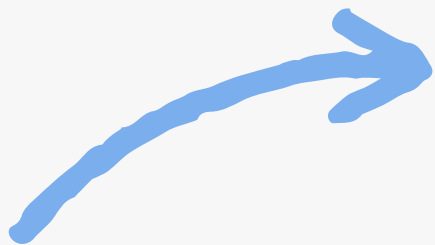


Effective TypeScript

Item 39.



any를 구체적으로 변형해서 사용하기



@Bori-github

목 차

01

any 타입

02

배열 타입 구체적으로 사용하기

03

객체 타입 구체적으로 사용하기

04

함수 타입 구체적으로 사용하기

05

마무리

O'REILLY®

Effective TypeScript

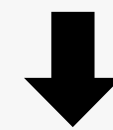
62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

자바스크립트에서 표현할 수 있는 모든 값을 아우르는 큰 범위의 타입

- 모든 숫자, 문자열, 배열, 객체, 정규식, 함수, 클래스, DOM 엘리먼트, `null`, `undefined`를 포함
- 일반적으로 any 보다 더 구체적으로 표현할 수 있는 타입이 존재할 가능성이 높다.



타입의 안정성을 높이기 위해 보다 구체적인 타입을 사용해야 합니다.



```
1 function getLengthBad(array: any) {  
2     return array.length;  
3 }  
4  
5 function getLength(array: any[]) {  
6     return array.length;  
7 }
```

매개변수로 **any** 타입을 사용한 것과 **any[]** 타입을 사용한 것의 차이점은 무엇일까요?

배열 타입 구체적으로 사용하기

✓ 매개변수로 **any[]** 타입을 사용할 때의 장점

```
1 function getLengthBad(array: any) {  
2   return array.length;  
3 }  
4  
5 function getLength(array: any[]) {  
6   return array.length;  
7 }
```

Annotations in the image:

- For line 1: `function getLengthBad(array: any): any`
- For line 2: `any` (pointing to `array.length`)
- For line 4: `function getLength(array: any[]): number`
- For line 6: `(property) Array<any>.length: number` (pointing to `array.length`)

- 매개변수가 배열인지 체크
- 함수 내의 `array.length` 타입 체크
- 함수의 반환 타입이 `any` 대신 `number`로 추론

➡ **타입의 안정성을 높임**

객체 타입 구체적으로 사용하기

만약 함수의 매개변수가 객체이지만 **값**을 알 수 없다면?

```
1 function hasTwelveLetterKey(o: object) {  
2   for (const key in o) {  
3     if (key.length === 12) {  
4       console.log(key, o[key]);  
5       return true;  
6     }  
7   }  
8   return false;  
9 }
```

'string' 형식의 식을 '{}' 인덱스 형식에 사용할 수 없으므로 요소에 암시적으로 'any' 형식이 있습니다.
'{' 형식에서 'string' 형식의 매개 변수가 포함된 인덱스 시그니처를 찾을 수 없습니다. ts(7053)

```
const key: string
```

object 타입 : 객체의 키를 열거할 수 있으나 속성에 접근할 수 없다.

객체의 속성에 접근해야 한다면?



```
1 function hasTwelveLetterKey(o: {[key: string]: any}) {  
2     for (const key in o) {  
3         if (key.length === 12) {  
4             console.log(key, o[key]); // 에러가 발생하지 않습니다.  
5             return true;  
6         }  
7     }  
8     return false;  
9 }
```


함수 타입 구체적으로 사용하기

✅ 함수의 타입을 구체화할 수 있는 세 가지 방법



```
1 // 매개변수 없이 호출 가능한 모든 함수
2 type Fn = () => any;
3
4 // 매개변수가 1개인 함수
5 type FnWithArg = (arg: any) => any;
6
7 // 매개변수가 여러 개인 함수
8 type FnWithArgs = (...arg: any[]) => any;
```


- any 타입은 모든 타입을 허용하기 때문에 사용하기 전에 정말로 모든 값이 허용되어야 하는지 검토해야 합니다.
- 타입의 안정성을 높이기 위해 단순한 any 타입을 사용하는 것보다 예제와 같이 **구체적인 형태를 사용**해야 합니다.



Thanks!

Do you have any questions?

qhflrnfl4324@gamil.com

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

