

Item 31.

타입 주변에 null 값 배치하기

Effective TypeScript

# 목 차

1. strictNullChecks 란?
2. strictNullChecks 옵션 사용하기
3. 값이 전부 null 이거나 전부 null 이 아닌 경우로 구분하기
4. 마무리

O'REILLY®

# Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

# 1. strictNullChecks 란?

## null과 undefined 값을 엄격하게 다루는 TypeScript 컴파일러의 옵션

### ✓ 타입 안정성이 향상

변수 또는 매개변수에 null 또는 undefined를 할당 불가하여 값이 예상되는 곳에서 오류를 사전에 발견

### ✓ 실수 방지

null 또는 undefined를 실수로 다른 값과 혼동하여 사용하는 것을 방지

### ✓ 코드의 명확성

null과 undefined의 사용을 명시적으로 처리하므로 코드의 의도를 명확히 파악 가능

## 2. strictNullChecks 옵션 사용하기

함수 extent : 숫자들의 최소값과 최대값을 계산

```
1 function extent(nums: number[]) {
2   let min, max;
3   for (const num of nums) {
4     if (!min) {
5       min = num;
6       max = num;
7     } else {
8       min = Math.min(min, num);
9       max = Math.max(max, num);
10    }
11  }
12  return [min, max];
13 }
```

(local function) extent(nums: number[]): (number | undefined)[]

① let min: any let max: any


②, ③ let min: number | undefined

④ 'number | undefined' 형식의 인수는 'number' 형식의 매개 변수에 할당될 수 없습니다.  
'undefined' 형식은 'number' 형식에 할당할 수 없습니다. ts(2345)  
let max: number | undefined

## 2. strictNullChecks 옵션 사용하기

### 문제 해결 방법

- 변수에 타입을 명시적으로 지정하거나 초기값 할당하기



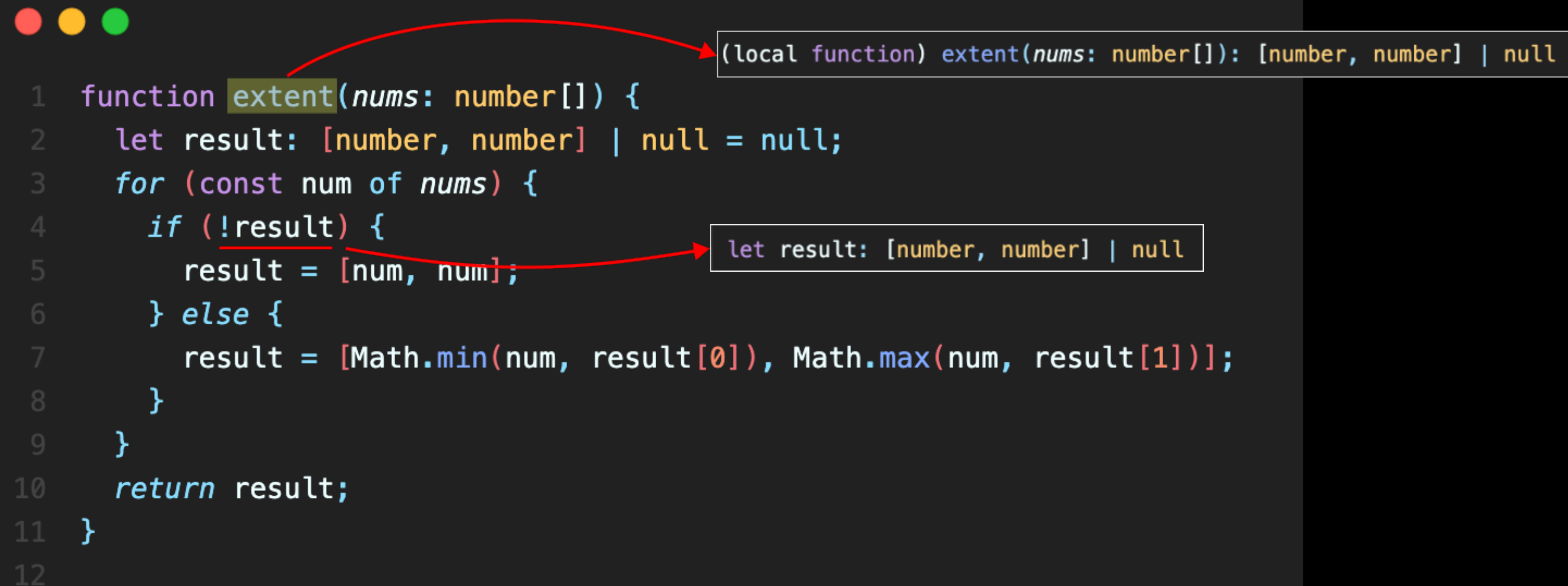
```
1 // 타입을 명시적으로 지정
2 let min: number | null;
3 let max: number | null;
4
5 // 변수 초기화
6 let min = nums[0];
7 let max = nums[0];
```

⇒ 각 변수에 대한 체크가 필요!



### 3. 값이 전부 null 이거나 전부 null 이 아닌 경우로 구분하기

두 변수를 하나의 객체에 넣어 null 체크하기



```
1 function extent(nums: number[]) {
2   let result: [number, number] | null = null;
3   for (const num of nums) {
4     if (!result) {
5       result = [num, num];
6     } else {
7       result = [Math.min(num, result[0]), Math.max(num, result[1])];
8     }
9   }
10  return result;
11 }
12
```

(local function) extent(nums: number[]): [number, number] | null

let result: [number, number] | null

⇒ 함수의 반환값을 단일 객체를 사용하여 설계를 개선하고,  
타입스크립트가 null 값 사이의 관계를 이해할 수 있도록하여 버그 제거

## 4. 마무리

- **strictNullChecks** 옵션을 사용하면 null 값과 관련된 문제점을 찾을 수 있고, null 값에 대한 처리를 명확하게 할 수 있어 코드의 안정성을 높일 수 있습니다.
- 코드 설계 시 한 값의 null 여부가 다른 값의 null 여부를 암시적으로 관련되도록 설계하면 안됩니다.
- 변수의 타입을 지정하고, 초기값을 할당하여 null 체크를 명시적으로 할 수 있도록 합니다.
- 여러 개의 변수를 **하나의 객체**에 넣어 사용하면 코드가 간결해지고 null 체크가 용이해집니다.

감사합니다 :)

Effective TypeScript

@Bori-github(이보리)