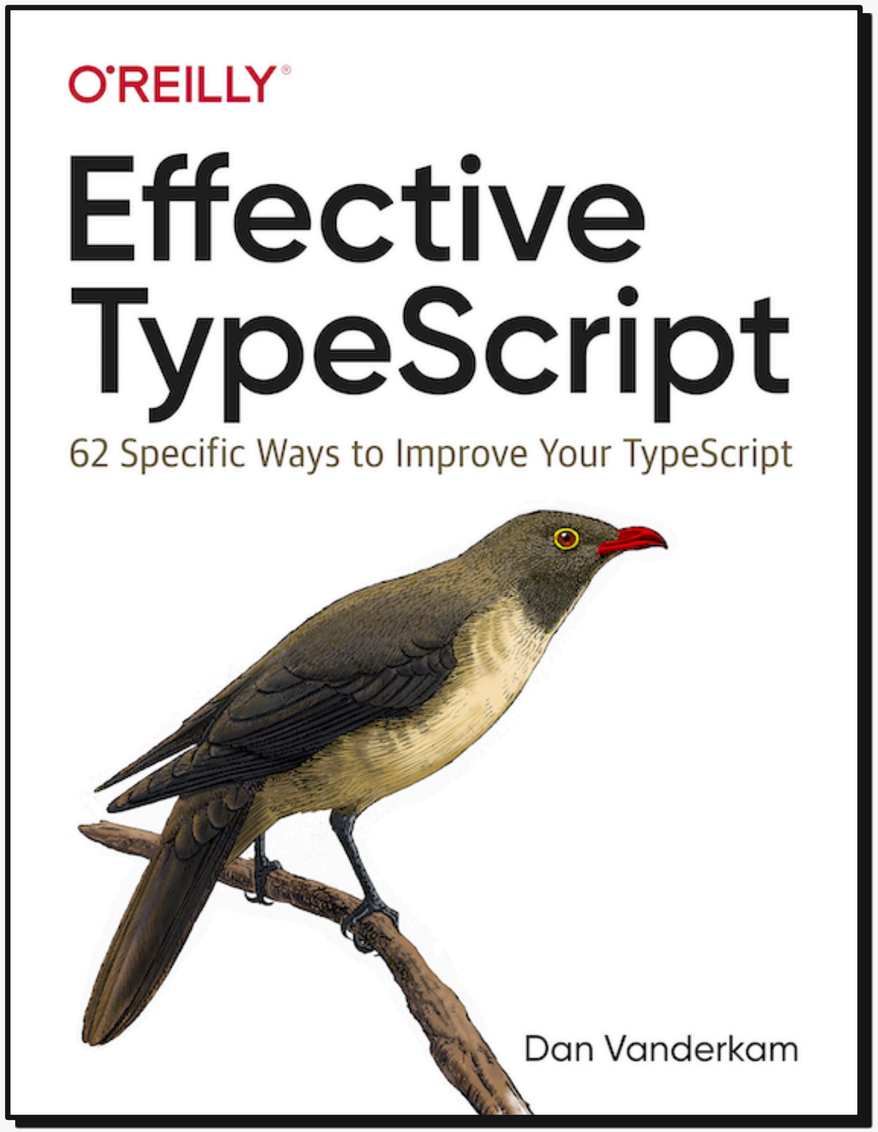
Effective TypeScript

Item 43.

몽키 패치보다는 안전한 타입을 사용하기

@Bori-github

목 차



01 몽키 패치란?

런타임 중인 프로그램의 내용이 변경되는 것

- JavaScript는 객체와 클래스에 임의의 속성을 추가할 수 있을 만큼 유연

- 객체의 메소드를 선언시점이 아닌 사용시점에 변경해서 사용 가능

몽키 패치 예시 코드 :

```
console.log_changed = console.log;
console.log = (arguments) => {console.log_changed(`몽키 패치 테스트 : ${arguments}`)};

console.log('테스트 완료');
```

몽키 패치 예시 결과 :

```
> console.log_changed = console.log
  console.log = (arguments) => {console.log_changed(`몽키 패치 테스트 : ${arguments}`)}

  console.log('테스트 완료')
  몽키 패치 테스트 : 테스트 완료                                    VM545:2
```

01 몽키 패치란?

- ☑ window나 document에 값을 할당하여 전역 변수 만들기

```
document.monkey_patch = 'Tada!'
```

- ☑ DOM 요소에 데이터를 추가하기

```
const el = document.getElementById('monkey_patch');
el.result = 'Tada!'
```

⇒ 전역 변수의 사이드 이펙트 고려

- ☑ 내장 기능의 프로토타입에 속성 추가하기

```
> RegExp.prototype.monkey_patch = 'Tada!';
'Tada!'
> /123/.monkey_patch
'Tada!'
```

⇒ 임의의 정규식에도 속성 추가

02 몽키 패치에 타입스크립트 더하기

☑️ document 객체에 속성을 추가하여 전역 변수 생성

```
1  document.monkey_patch = 'Tada!'
```

```
'Document' 형식에 'monkey_patch' 속성이 없습니다. ts(2339)
any
```

🚫 문제점: 타입 체커가 임의로 추가한 속성에 대해서는 알지 못하여 에러 발생

02 몽키 패치에 타입스크립트 더하기

☑ any 단언문 사용하기

```
1  (document as any).monkey_patch = 'Tada!';  // 정상
2
3  (document as any).monkey_p = 'Tada!';  // 오타가 발생했으나 정상
4  (document as any).monkey_patch = /Tada!/;  // 타입이 다르지만 정상
```

⬇

any의 사용으로 타입의 안정성이 낮아지고,

오타가 발생해도 타입 체커가 **오류를 발생시키지 않음**

03 몽키 패치보다는 안전한 타입을 사용하기

✅ interface의 보강(augmentation) 사용하기

```ts
interface Document {
  /** 몽키 패치 속성 추가 */
  monkey_patch: string;
}

document.monkey_patch = 'Tada!';  // 정상
```

- 타입의 안정성을 높임
- 속성에 주석 사용 가능
- 속성에 자동완성 기능
- 몽키 패치가 적용된 정확한 기록

🚫 주의 사항

- 보강은 전역적으로 적용 ⇒ 코드를 분리할 수 없다.
- 애플리케이션이 실행되는 동안 속성을 할당한 경우 실행 시점에서 보강 적용 불가

03 몽키 패치보다는 안전한 타입을 사용하기

✅ 더 구체적인 타입 단언문 사용하기

```ts
interface MonkeyPatchDocument extends Document {
  monkey_patch: string;
}

(document as MonkeyPatchDocument).monkey_patch = 'Tada!';
```

Document 타입 확장을 통해 새로운 타입을 생성하여 타입 단언을 적용

⬇

전역적으로 적용되는 문제를 해결하여 import 한 곳에서만 사용 가능

05

마무리

- 몽키 패치를 남용하지 않도록 구조를 설계하는 것이 좋습니다.

(몽키 패치의 좋은 사례 : Polyfill은 구형 브라우저에서 모던 자바스크립트의 기능을 사용할 수 있게 몽키패칭 함)

- 내장 타입에 데이터를 저장해야 하는 경우, 보강 기능을 이용하거나 내장 타입을 확장하여 타입 단언을 적용하는 방법과 같은 안전한 타입 접근법을 사용해야 합니다.

- 보강 기능을 사용할 경우 모듈의 영역 문제를 이해하고 사용해야 합니다.

Thanks!

Do you have any questions?

qhflrnfl4324@gamil.com
https://github.com/Bori-github