

Item 3.

코드 생성과 타입이 관계 없음을 이해하기

Effective TypeScript

목 차

1. 타입스크립트 컴파일러의 역할
2. 타입 오류가 있는 코드도 컴파일이 가능합니다.
3. 런타임에는 타입 체크가 불가능합니다.
4. 타입 연산은 런타임에 영향을 주지 않습니다.
5. 런타임 타입은 선언된 타입과 다를 수 있습니다.
6. 타입스크립트 타입으로는 함수를 오버로드 할 수 없습니다.
7. 타입스크립트 타입은 런타임 성능에 영향을 주지 않습니다.

O'REILLY®

Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

1. 타입스크립트 컴파일러의 역할

1. 최신 타입스크립트/자바스크립트를 구버전의 자바스크립트로 **트랜스파일**
2. 코드의 **타입 오류를 체크**

위의 두 가지 역할은 서로 완벽히 독립적!

타입스크립트가 자바스크립트로 변환될 때, 자바스크립트 실행 시점에 타입은 영향을 미치지 않는다.

2. 타입 오류가 있는 코드도 컴파일이 가능합니다.

컴파일은 타입 체크와 **독립적**으로 동작

⇒ 타입 오류가 있는 코드도 컴파일이 가능

- 타입스크립트 오류는 문제가 될 만한 부분을 알려주는 것이므로, 빌드를 멈추지는 않는다.
(반면, C나 자바 같은 언어는 타입 체크와 컴파일이 동시에 이루어짐)
- tsconfig.json에 noEmitOnError 설정하면, 오류가 있을 때 컴파일하지 않는다.

3. 런타임에는 타입 체크가 불가능합니다.

타입스크립트의 타입은 **제거 가능(erasable)**

⇒ 자바스크립트로 컴파일 되는 과정에서 모든 인터페이스, 타입, 타입 구문 제거

런타임에 타입 정보를 유지하는 방법

1. 속성의 존재 여부 체크

⇒ 속성 체크는 런타임에 접근 가능한 값

2. 타입 정보를 명시적으로 저장하는 태그 기법 사용

⇒ 타입 정보를 명시적으로 저장하여 해당 값을 통해 타입 정보를 유지

3. 클래스를 이용한 타입 선언

⇒ 클래스로 선언하면 타입과 값으로 모두 사용 가능

4. 타입 연산은 런타임에 영향을 주지 않습니다.

문제점

⇒ 자바스크립트로 컴파일되는 과정에서 **타입 연산 제거**



```
1 function asNumber(val: number | string): number {  
2     return val as number;  
3 }
```



```
1 // 자바스크립트 코드로 변환  
2 function asNumber(val) {  
3     return val;  
4 }
```

4. 타입 연산은 런타임에 영향을 주지 않습니다.

문제 해결 방법

- 런타임의 타입을 체크
- 자바스크립트 연산을 통해 변환을 수행



```
1 function asNumber(val: number | string): number {  
2   return typeof(val) === 'string' ? Number(val) : val;  
3 }
```


5. 런타임 타입은 선언된 타입과 다를 수 있습니다.



```
1 function setLightSwitch(value: boolean) {  
2   switch (value) {  
3     case true:  
4       turnLightOn();  
5       break;  
6     case false:  
7       turnLightOff();  
8       break;  
9     default:  
10      console.log(`I'm afraid I can't do that.`);  
11   }  
12 }
```

: boolean 은 타입 선언문

⇒ 런타임에 제거

 만약, 네트워크 호출로부터 받아온 값으로 함수를 실행하는 경우

- 요청의 결과로 boolean 값을 반환하라고 선언하더라도 실제로 그렇게 되리라는 보장이 없다.

6. 타입스크립트 타입으로는 함수를 오버로드할 수 없습니다.

함수 오버로딩?

- 같은 이름의 함수에 매개 변수를 다르게 사용하여 매개 변수에 따라 다른 함수가 실행되는 것을 의미
- 자바스크립트에는 함수 오버로딩 기능이 없다.
- 타입과 런타임의 동작이 무관하므로, 타입스크립트도 함수 오버로딩이 불가

하지만,

타입스크립트는 **함수 오버로딩 기능을 지원**

6. 타입스크립트 타입으로는 함수를 오버로드할 수 없습니다.

타입스크립트의 함수 오버로딩

- 선언부(오버로드 시그니처)와 구현부(구현 시그니처)로 나뉨
 - 선언부 : 매개 변수의 타입 지정
 - 구현부 : 함수 구현
- 하나의 함수에 여러 선언문을 작성할 수 있으나, 구현체는 오직 하나

7. 타입스크립트 타입은 런타임 성능에 영향을 주지 않습니다.

1. 타입과 타입 연산자는 자바스크립트 변환 시점에 제거되기 때문에, **런타임의 성능에 아무런 영향을 주지 않음**

⇒ 타입스크립트 컴파일러는 런타임 오버헤드가 없다.

2. 대신, **빌드타임 오버헤드**가 있다.

감사합니다 :))

Effective TypeScript

@Bori-github(이보리)