

3장 타입의 모든 것

*타입

값과 이 값으로 할 수 있는 일의 집합

타입을 알면 어떤 일을 할 수 있는지 알 수 있다.

타입스크립트는 특정 타입만 와야 할 때, 이를 명시할 수 있는 언어

타입 추론 vs 타입 어노테이션

타입의 가나다

any

- 꼭 필요한 상황이 아니라면 사용하지 말자
- any를 사용하면 자바스크립트처럼 동작하기에 타입스크립트 사용 이유가 없어진다.

unknown

- 타입을 미리 알 수 없는 어떤 값이 있다면, any 대신 unknown을 사용하도록 하자.

boolean

- true, false 두 가지 값을 가진다. (비교 연산 가능)

타입 리터럴

값을 한정하여 안전성을 확보해주는 기능

number

- 모든 숫자(정수, 소수, 양수, 음수, Infinity, NaN 등)의 집합이며 숫자 관련 연산을 수행할 수 있다.
- 긴 숫자를 처리할 때, 숫자 분리자를 이용해 숫자를 읽기 쉽게 만들 수 있다.

```
let oneMillion = 1_000_000;
```

bigint

- 자바스크립트와 타입스크립트에 새로 추가된 타입으로 이를 이용하면 라운딩 관련 에러 걱정 없이 큰 정수를 처리할 수 있다.
- number 타입은 최대 $2^{53}-1$ 까지 표현 가능하지만 bigint는 이보다 큰 수도 표현이 가능하다. 모든 숫자 관련 연산도 가능하다.

```
let a = 1234n; // bigint
```

string

- 모든 문자열의 집합으로 연결(+), 슬라이스(slice) 등의 연산을 수행할 수 있다

symbol

- 심벌 키를 사용하면 사람들이 잘 알려진 키만 사용하도록 강제할 수 있으므로 키를 잘못 설정하는 실수를 방지한다

객체

- object는 any보다 조금 더 좁은 타입으로 다음과같이 명시적으로 선언할 수 있다.

```
let a: object = {  
  b: 'x'  
}
```

타입 별칭

- 타입 별칭으로 타입 가리킬 수 있다.
- 하나의 타입을 두 번 정의할 수는 없다.
- 블록 영역에 적용된다.

유니온과 인터섹션

- 집합처럼 연산 수행이 가능하다.

배열

- 연결(concatenation), 푸시(push), 검색(searching), 슬라이스(slicing) 등을 지원하는 특별한 객체다.

튜플

- 배열의 서브타입이다.
- 길이가 고정
- readonly 배열은 명시적 타입 어노테이션으로 만들 수 있다.

null, undefined, void, never

- null과 undefined는 타입스크립트에서 값이 유일해 특별한 타입이다.
- void는 명시적으로 아무것도 반환하지 않는 함수의 반환 타입
- never는 절대 반환하지 않는(예외를 던지거나 영원히 실행되는) 함수 타입

열거형

- 해당 타입으로 사용할 수 있는 값을 열거하는 기법
- const enum은 더 안전한 접근 허용(역방향 찾기 지원하지 않음)