

 현재 문서: 5.6.2 Type

- 1 타입의 종류
- 2 타입 시스템
- 3 오류의 종류
 - 3.1 구문 오류
 - 3.2 타입 오류
- 4 할당 가능성 (assignability)
- 5 타입 어노테이션 (type annotation)
- 6 타입 형태

1 타입의 종류

‘타입’은 자바스크립트에서 다루는 값의 형태에 대한 설명입니다. 여기서 ‘형태’란 값에 존재하는 속성과 메서드 그리고 내장되어 있는 `typeof` 연산자가 설명하는 것을 의미합니다. 타입스크립트의 가장 기본적인 타입은 자바스크립트의 일곱 가지 기본 원시 타입과 동일합니다.

```

null // null
undefined // undefined
true // boolean
"Louise" // string
1337 // number
1377n; // bigint
Symbol("Frnklin") // symbol
```

2 타입 시스템

타입 시스템은 프로그래밍 언어가 프로그램에서 가질 수 있는 타입을 이해하는 방법에 대한 규칙 집합입니다. 타입 스크립트의 타입 시스템은 다음과 같이 동작합니다.

- 1. 코드를 읽고 존재하는 모든 타입과 값을 이해합니다.
- 2. 각 값이 초기 선언에서 가질 수 있는 타입을 확인합니다.
- 3. 각 값이 추후 코드에서 어떻게 사용 될 수 있는지 모든 방법을 확인합니다.
- 4. 값의 사용법이 타입과 일치하지 않으면 사용자에게 오류를 표시합니다.

3 오류의 종류

- 구문 오류: 타입스크립트가 자바스크립트로 변환되는 것을 차단한 경우
- 타입 오류: 타입 검사기에 따라 일치하지 않는 것이 감지된 경우

3.1 구문 오류

구문 오류는 타입스크립트가 코드로 이해할 수 없는 잘못된 구문을 감지할 때 발생합니다. 이는 타입스크립트가 타입스크립트 파일에서 자바스크립트 파일을 올바르게 생성할 수 없도록 차단합니다.

[blocked URL](#)

3.2 타입 오류

타입 오류는 타입스크립트의 타입 검사기가 프로그램의 타입에서 오류를 감지했을 때 발생합니다. 오류가 발생했다고 해서 타입스크립트 구문이 자바스크립트로 변환되는 것을 차단하지는 않습니다. 하지만 코드가 실행되면 무언가 충돌하거나 예기치 않게 작동할 수 있음을 나타냅니다.

[blocked URL](#)

4 할당 가능성 (assignability)

타입스크립트는 변수의 초기값을 읽고 해당 변수가 허용되는 타입을 결정합니다.
나중에 해당 변수에 새로운 값이 할당되면, 새롭게 할당된 값의 타입이 변수의 타입과 동일한지 확인합니다.

```
let lastName = "king";  
lastName = true;
```

[blocked URL](#)

5 타입 어노테이션 (type annotation)

때로는 변수에 타입스크립트가 읽어야 할 초기값이 없는 경우도 있습니다.
타입스크립트는 나중에 사용할 변수의 초기 타입을 파악하려고 시도하지 않습니다.
그리고 기본적으로 변수를 암묵적인 any 타입으로 간주합니다.
즉, 변수는 세상의 모든 것이 될 수 있음을 나타냅니다.
타입스크립트는 초기값을 할당하지 않고도 변수의 타입을 선언할 수 있는 구문인 타입 애너테이션을 제공합니다.
타입 애너테이션은 변수 이름 뒤에 배치되며 콜론(:)과 타입이름을 차례대로 기재합니다.

```
let rocker: string;  
rocker = "Joan Jett";
```

[blocked URL](#)

6 타입 형태

타입스크립트는 변수에 할당된 값이 원래 타입과 일치하는지 확인하는 것 이상을 수행합니다.
타입스크립트는 객체에 어떤 멤버 속성이 존재하는지 알고 있습니다.
만약 여러분이 코드에서 변수의 속성에 접근하려고 한다면 타입스크립트는 접근하려는 속성이 해당 변수의 타입에 존재하는지 확인합니다.

```
let rapper = "Queen Latifah";  
rapper.length;  
  
rapper.push('');
```

[blocked URL](#)

타입은 더 복잡한 형태, 특히 객체일 수도 있습니다.
다음 스택오버플로에서 타입스크립트는 cher 객체에 middleName 키가 없다는 것을 알고 오류를 표시합니다.

```
let cher = {  
  firstName: "Cherilyn",  
  lastName: "Sarkisian",  
};  
  
cher.middleName;
```

[blocked URL](#)