

# Ionic Study

Day 7

# 오늘 할 것들

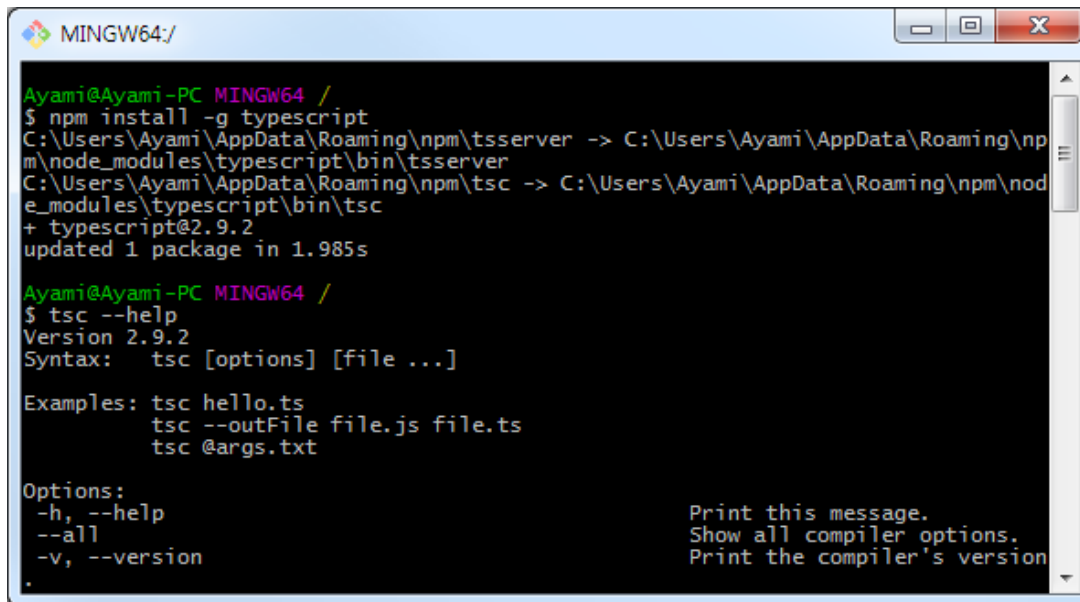
- 타입스크립트 알아보기

# 타입스크립트

- Typescript
  - Ionic3은 Angular와 타입스크립트에 기반을 두고 있음.
  - 타입스크립트는 컴파일러에 의해 자바스크립트로 변환되어 실제 동작.

# 타입스크립트

- Typescript 컴파일러 설치
  - npm install -g typescript



```
MINGW64:/
Ayami@Ayami-PC MINGW64 /
$ npm install -g typescript
C:\Users\Ayami\AppData\Roaming\npm\tsserver -> C:\Users\Ayami\AppData\Roaming\np
m\node_modules\typescript\bin\tsserver
C:\Users\Ayami\AppData\Roaming\npm\tsc -> C:\Users\Ayami\AppData\Roaming\npm\nod
e_modules\typescript\bin\tsc
+ typescript@2.9.2
updated 1 package in 1.985s

Ayami@Ayami-PC MINGW64 /
$ tsc --help
Version 2.9.2
Syntax:  tsc [options] [file ...]

Examples: tsc hello.ts
          tsc --outFile file.js file.ts
          tsc @args.txt

Options:
  -h, --help                Print this message.
  --all                     Show all compiler options.
  -v, --version             Print the compiler's version
  .
```

# 타입스크립트

- tsc 명령
  - 타입스크립트 컴파일러 호출
  - ts 확장자를 가진 타입스크립트 파일을 입력으로 주면 출력으로 동일 이름의 js 확장자를 가진 자바스크립트 파일이 생성됨.
  - node로 js파일 실행

# 타입스크립트

sample.ts

```
var message:string="Hello World";  
console.log(message)
```

컴파일



sample.js

```
var message="Hello World";  
console.log(message)
```

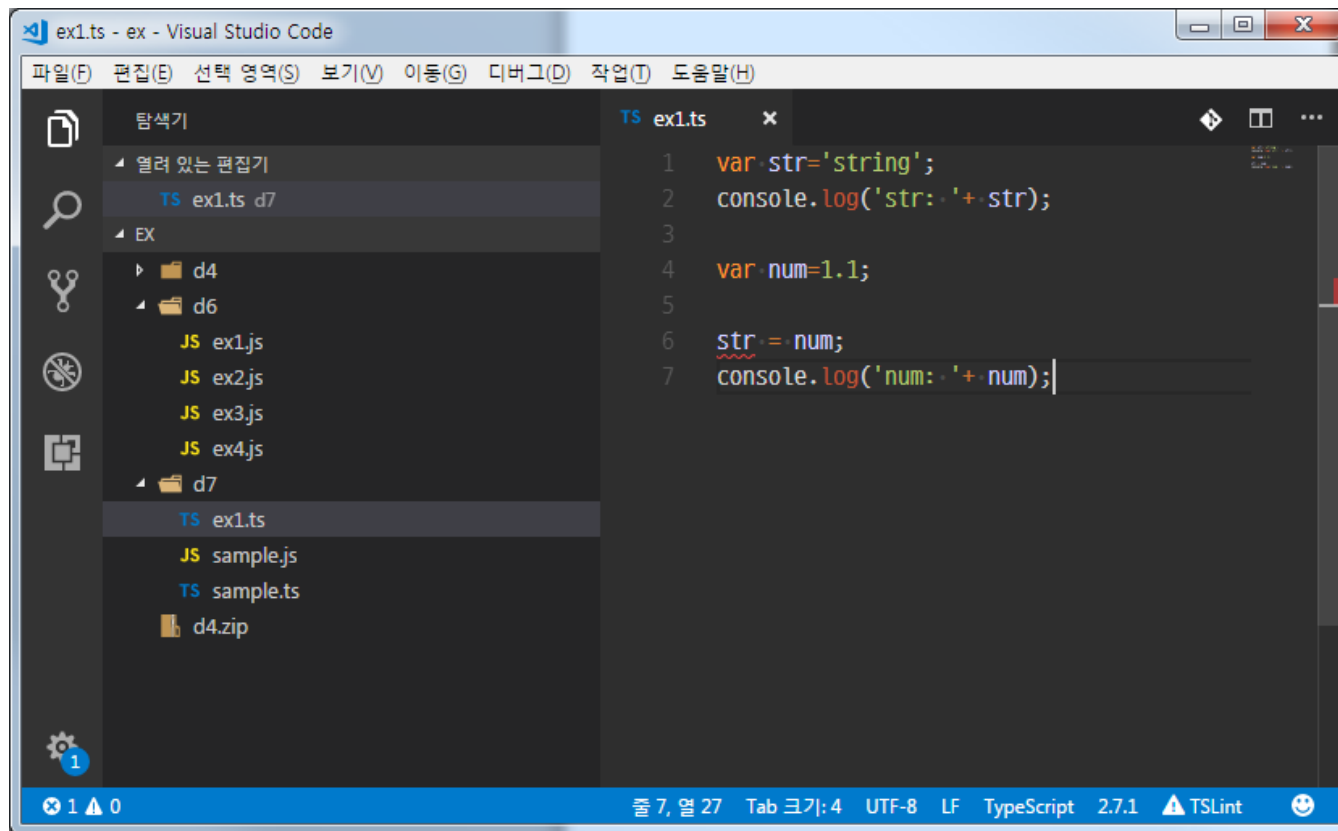
```
MINGW64:/c:/Users/Ayami/Desktop/??????/ex/d7  
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/ex/d7  
$ ls  
sample.ts  
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/ex/d7  
$ tsc sample.ts  
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/ex/d7  
$ ls  
sample.js sample.ts  
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/ex/d7  
$ node sample.js  
Hello World  
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/ex/d7  
$
```

# 타입스크립트

- 변수의 자료형
  - 타입스크립트에서는 변수에 변수의 형태(자료형)를 지정하고 동일 형태로만 사용 가능.
  - 자바스크립트는 변수의 형태를 지정할 필요가 없음 (하나의 변수를 여러 타입으로 사용 가능)

# 타입스크립트

- ex1.ts





# 타입스크립트

- 타입스크립트에서 변수의 선언
  - 변수 선언부에 사용할 형태를 알려준다
  - ex) `var variable:string='string'`
    - variable 변수는 string형을 가짐
  - ex) `var num:number=1.1`
    - num 변수는 number형을 가짐

# 타입스크립트

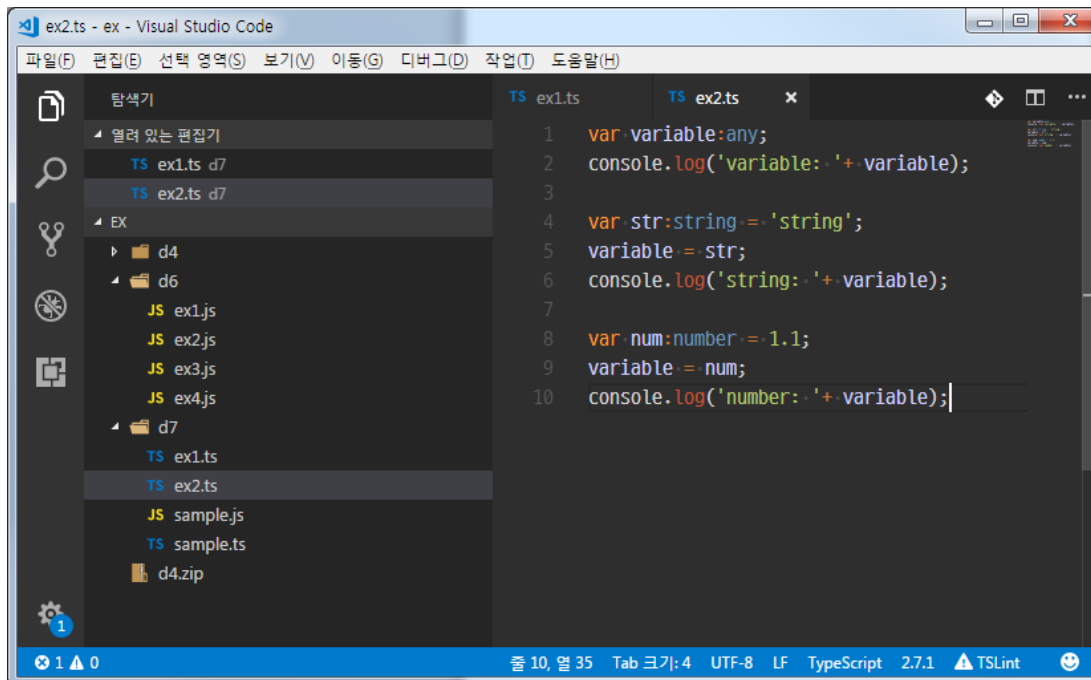
- 타입스크립트에서 지원하는 기본 자료형

타입	설명
number	64비트의 크기를 갖는 실수(정수+소수를 포함한)형 숫자
string	유니코드 문자열
boolean	true와 false의 값을 가지는 논리형
array<Type>	Type형태를 갖는 집합
union	하나의 변수가 여러 형태의 자료형을 가질 경우 사용
tuple	여러 형태의 array
any	정의되지 않은 임의의 타입

– 초기화되지 않은 변수는 undefined 값이 할당됨.

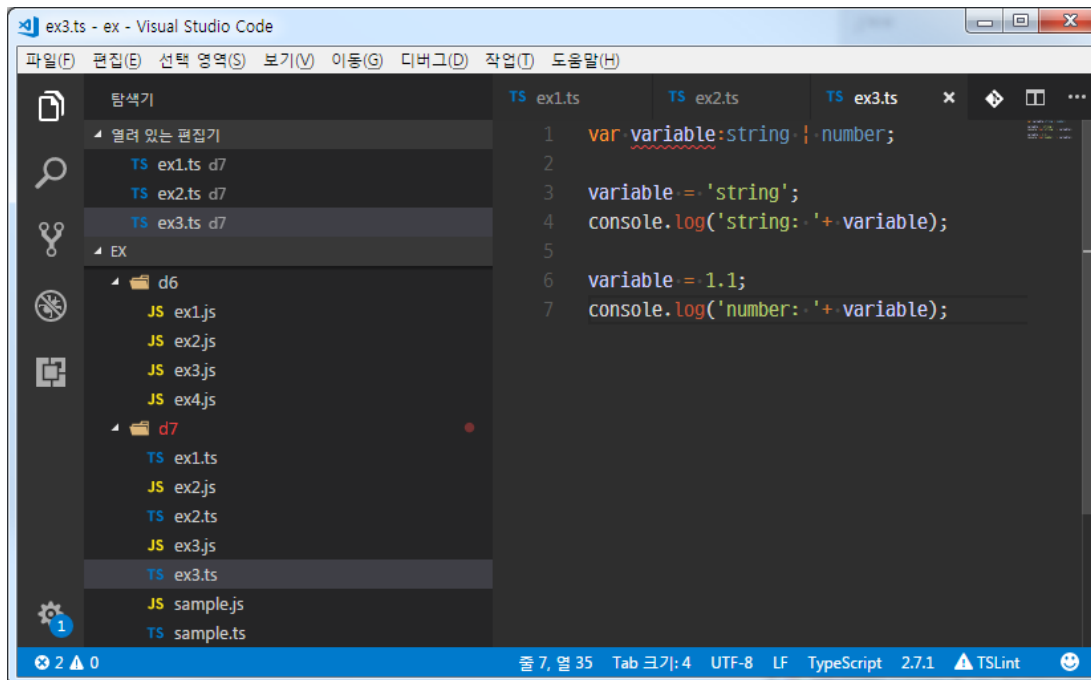
# 타입스크립트

- any 자료형 (ex2.ts)
  - 여러 타입으로 변수를 사용 가능.



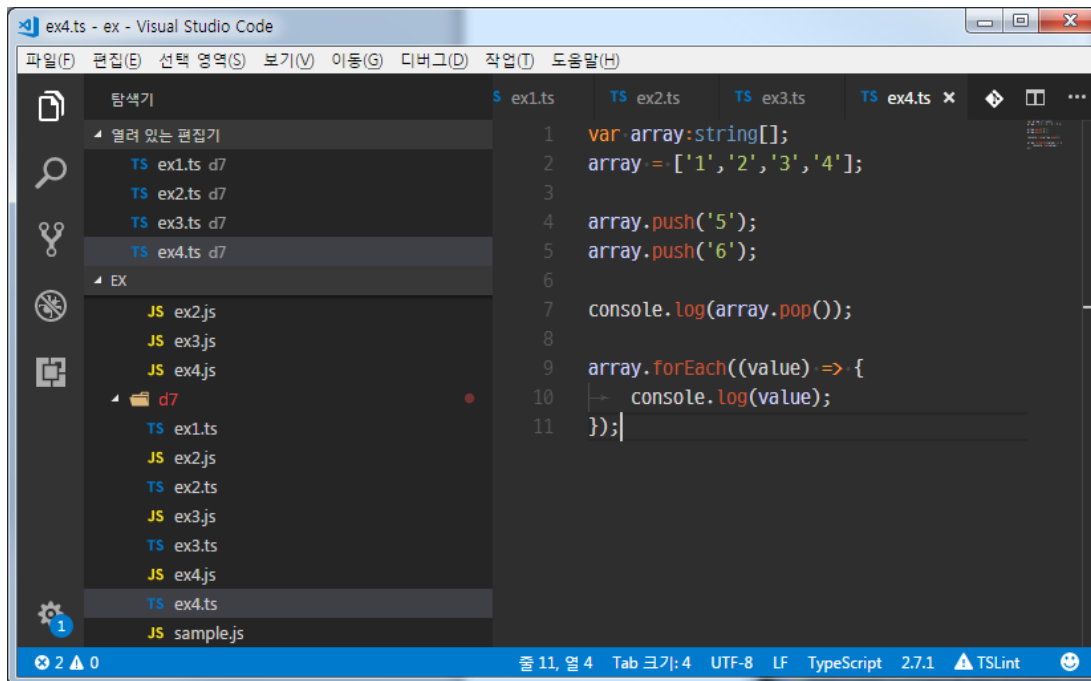
# 타입스크립트

- union 자료형 (ex3.ts)
  - 사용할 변수가 가지게 될 자료형을 미리 알고 있다면 any 대신 사용 가능. union은 각 형태의 or (|) 로 표기.



# 타입스크립트

- array 자료형 (ex4.ts)
  - 동일한 자료형의 집합은 array를 사용
  - 자바스크립트의 array 객체와 동일

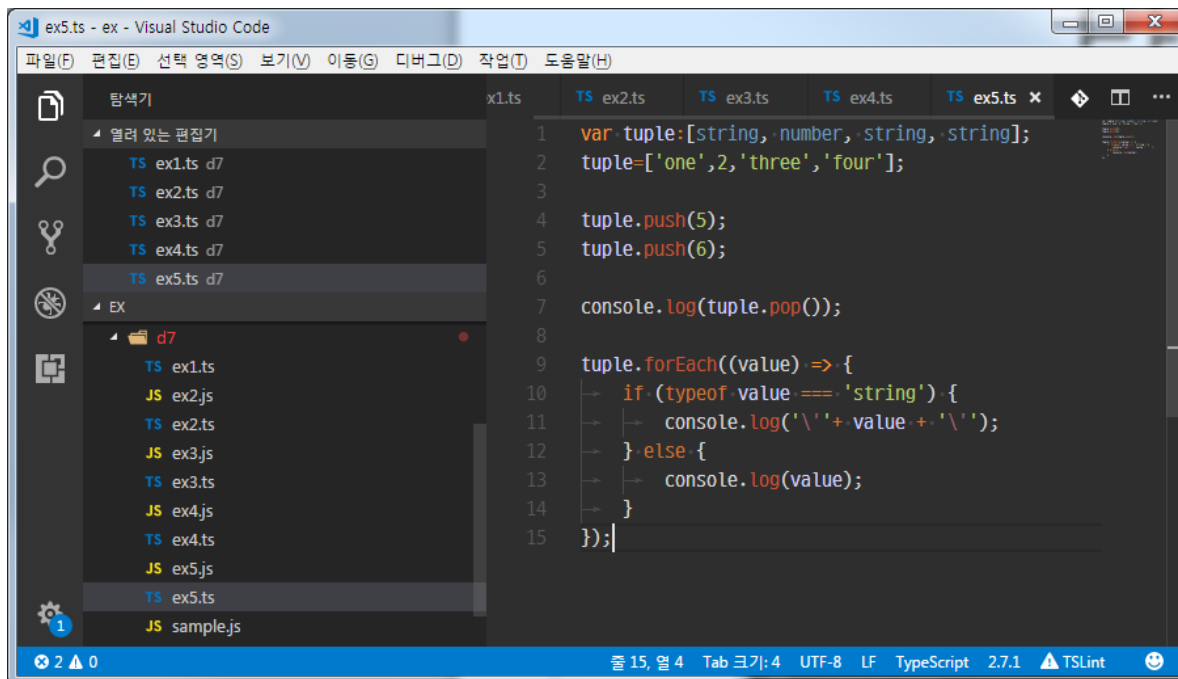


배열 리터럴을 이용한 방식  
`var array: string[];`

제네릭을 이용한 방식  
`var array: Array<string>`

# 타입스크립트

- tuple 자료형 (ex5.ts)
  - 자료형이 다른 여러 개의 변수는 튜플 타입으로 정의.
  - array 객체의 일부 메소드를 지원, 선언한 만큼 요소를 할당해야 함

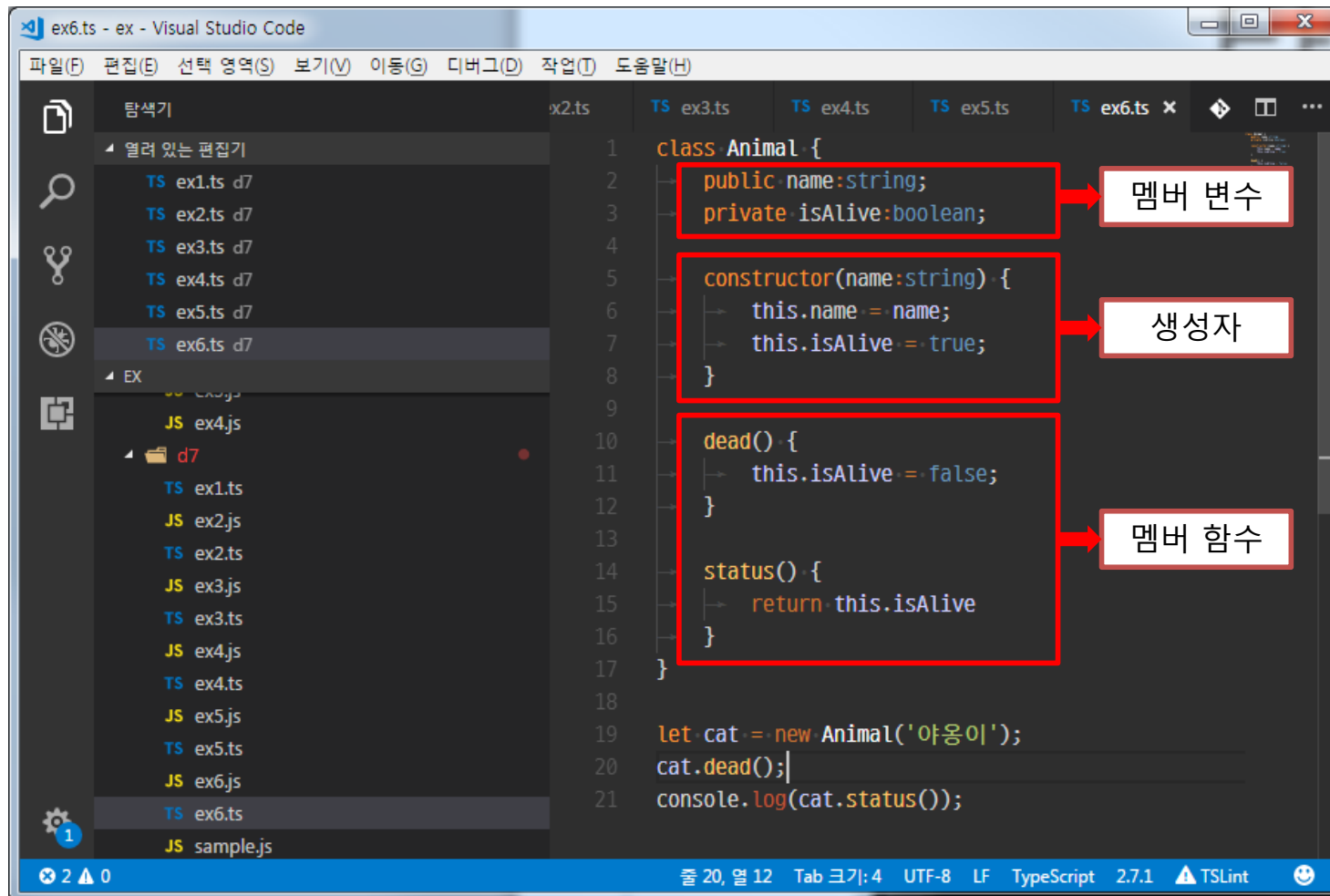


# 타입스크립트

- 사용자 정의 형태
  - 타입스크립트에서는 기본적으로 제공되는 자료형 외에 사용자가 원하는 형태를 클래스로 정의하여 사용하는 것이 가능.
  - 클래스는 생성자, 멤버 변수, 멤버 함수 세가지로 정의.
    - 생성자 : 클래스가 객체를 생성할 때 수행되는 함수
      - > 객체를 초기화하기 위한 작업을 수행

# 타입스크립트

- 클래스 (ex6.ts)





# 타입스크립트

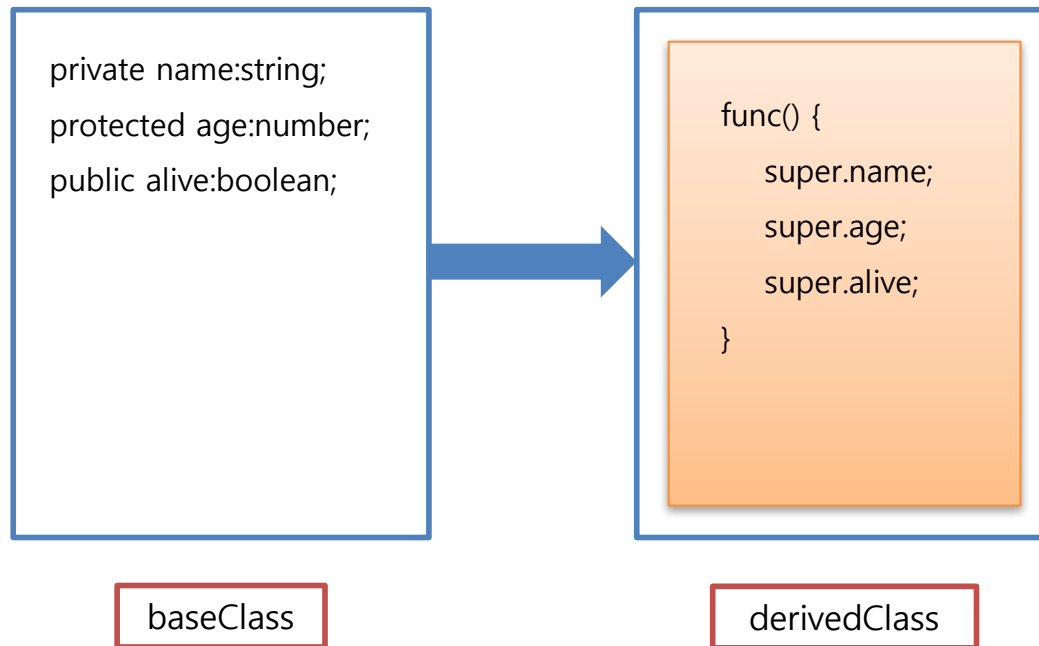
- 클래스
  - 접근 권한 : 멤버 변수 또는 멤버 함수에 접근할 수 있는 범위를 설정.
    - private : 동일한 클래스 내부에서만 접근 가능
    - protected : 동일한 클래스와 해당 클래스를 상속받은 클래스에서만 접근 가능
    - public : 동일 클래스 및 외부에서 접근 가능
  - this 키워드 : 클래스 내부의 필드에 접근하거나 함수를 호출 할 때 사용.

# 타입스크립트

- 클래스의 상속
  - 상속 : 원래 클래스의 멤버 변수와 멤버 함수를 새로운 클래스에서도 추가 선언 없이 사용하는 방법
    - 부모 클래스, 자식 클래스
      - 부모 클래스 : 특성을 제공할 클래스 (기본 클래스)
      - 자식 클래스 : 특성을 상속받을 클래스 (파생 클래스)
  - extends 키워드를 이용하여 클래스 상속 실현
  - 자식 클래스 내부에서는 super 키워드를 이용하여 부모 클래스의 멤버 함수를 호출할 수 있음.
    - protected, public 스코프만 가능.

# 타입스크립트

- 클래스



컴파일 가능 여부와 불가능일 경우 왜 그런건지?

# 타입스크립트

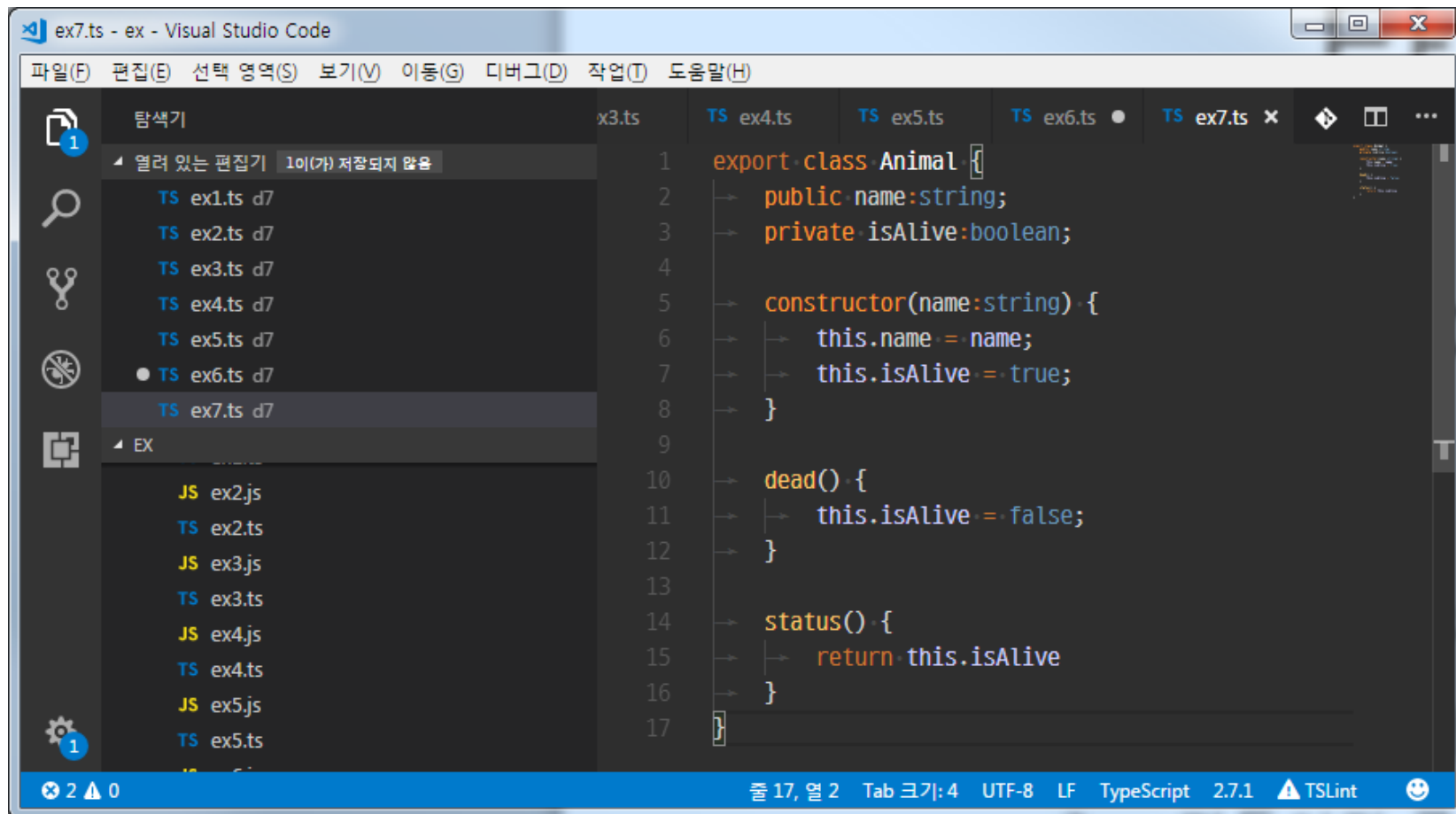
- 클래스의 모듈화

- 모듈화

- 만들어진 클래스를 다른 파일에서 사용하려면?
      1. 그 파일에서 또 클래스를 선언한다
      2. 만들어진 클래스를 어떻게든 불러와 재활용한다.
    - 같은걸 계속 만들어 내는것은 생산성 저하, 유지보수 최악의 특징. (원래 내용이 바뀌면 복사한 것의 내용도 바뀌야 하니깐)
    - 재활용을 위해 존재하는 키워드가 export 키워드임.
    - 내보낼 때는 클래스 선언부 앞에 export를 붙여주면 끝.

# 타입스크립트

- ex7.ts (ex6의 export 사용)

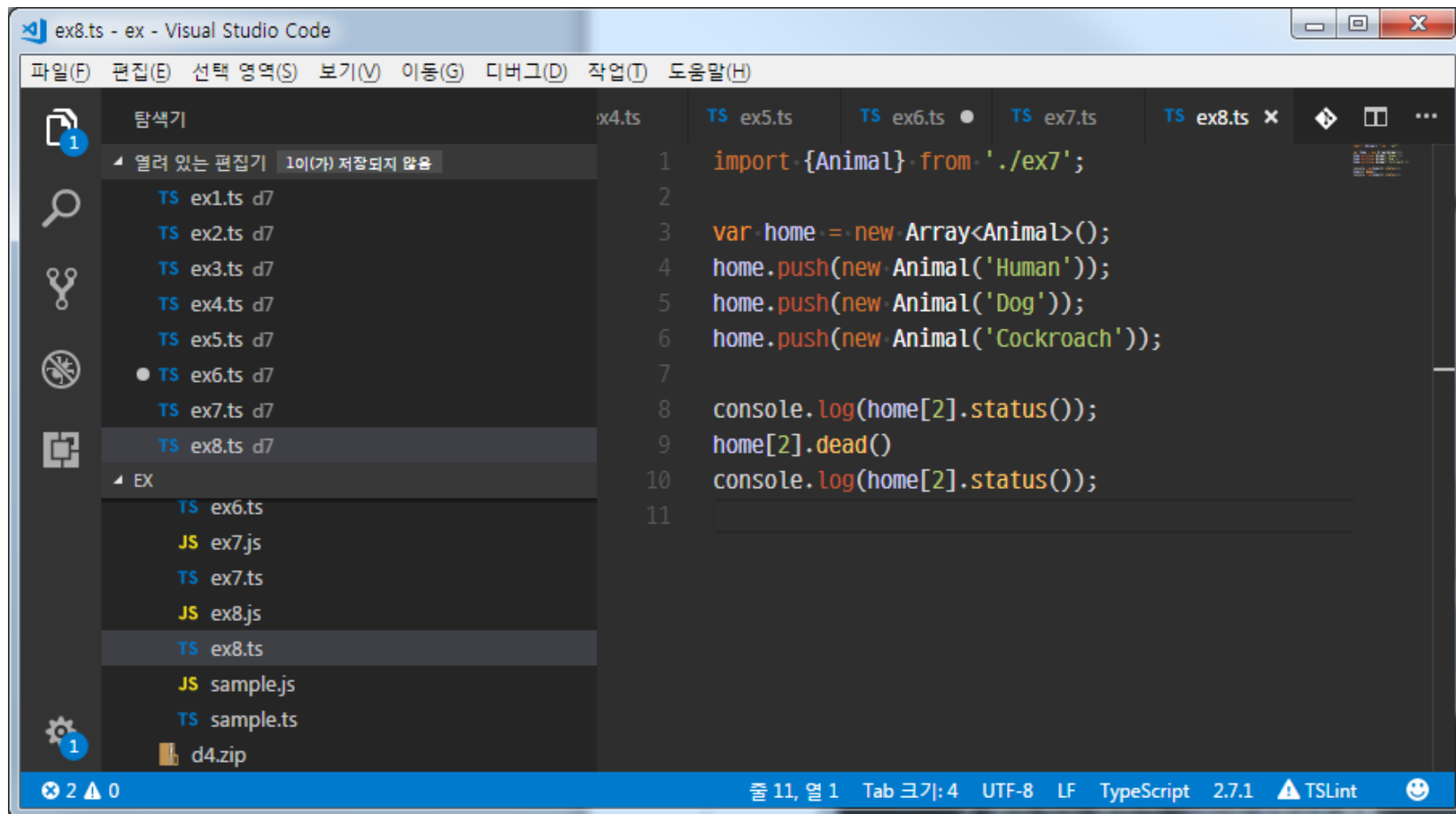


# 타입스크립트

- export로 내보낸 모듈 불러오기
  - import 키워드를 사용하여 export로 내보낸 모듈을 불러올 수 있음.
    - import {타입명} from '파일 경로';
  - 이렇게 불러온 모듈은 사용자 정의형으로 자유롭게 사용이 가능.

# 타입스크립트

- ex8.ts (바퀴벌레의 죽음)



오늘은 여기까지~

See you next day!