TypeScript 노트

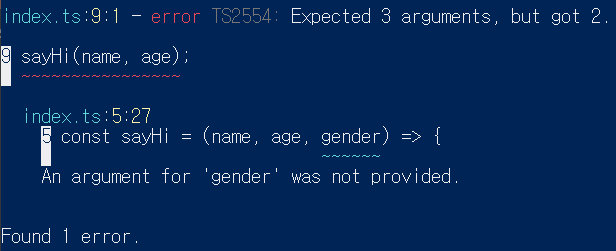
먼저 노마드코더 강좌 초반에서 타입스크립트 개발환경 세팅을 하고,

비디오 #4에서 tsc-watch세팅을 하자.

다음코드는 자바스크립트에선 정상적으로 작동하나, 타입스크립트에선 에러가 난다.



참고로, 맨 마지막 줄 export {}; 는 타입스크립트에서 관습적으로 반드시 들어가야 하는 코드다.



이는 매개변수가 하나 부족하기 때문. 즉, 타입스크립트를 쓰는 가장 큰 이유는, 멍청한 실수를 방지하기 위함이다.

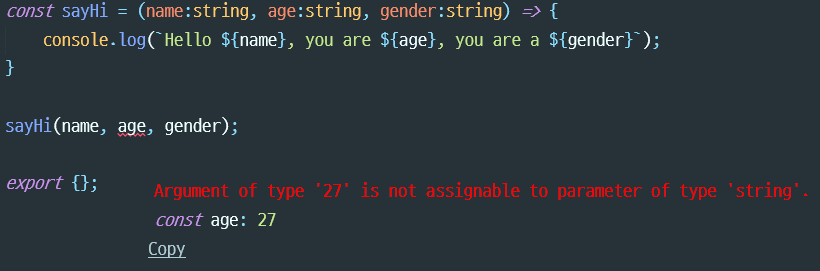
그러나 다음경우엔 정상작동한다.



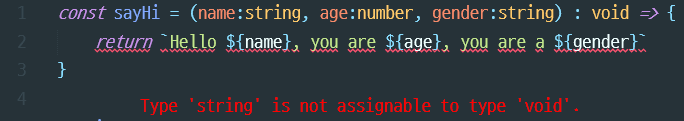


왜냐면, 이 경우엔 변수 뒤에 물음표를 붙여서 들어가도 되고 안 들어가도 된다고 ‘명시’해줬기 때문이다. 적어도, 우리가 명시했기 때문에 이렇게 돌아간다는 건 알 수 있다.

이제 가장 중요한 타입에 대해 말하자면,



매개변수의 타입을 지정해줬기 때문에 타입 불일치 에러가 잡힌다.

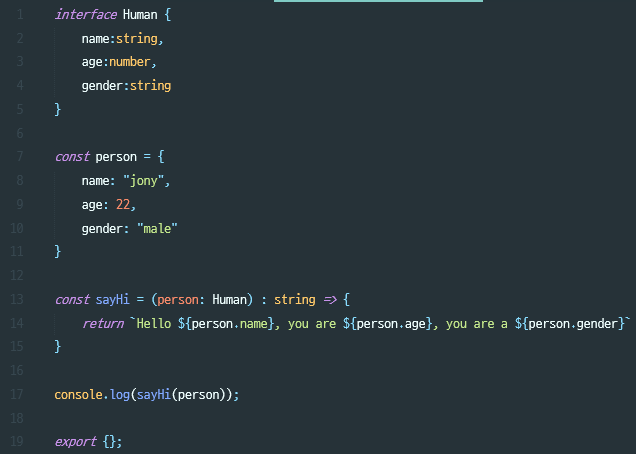


이 경우엔, 반환값이 string인데 void로 맞춰놓으니깐 에러가 뜬 것. 반환타입을 string으로 바꿔주면 아무 문제가 안 생긴다.

이제 object의 경우에, type을 어떻게 지정해서 쓸 수 있는가 알아보자.

이 경우엔 interface를 써야한다.

**interface**



person이라는, 하나의 객체를 Human이라는 사용자정의 자료형으로 바꾼 것이다!

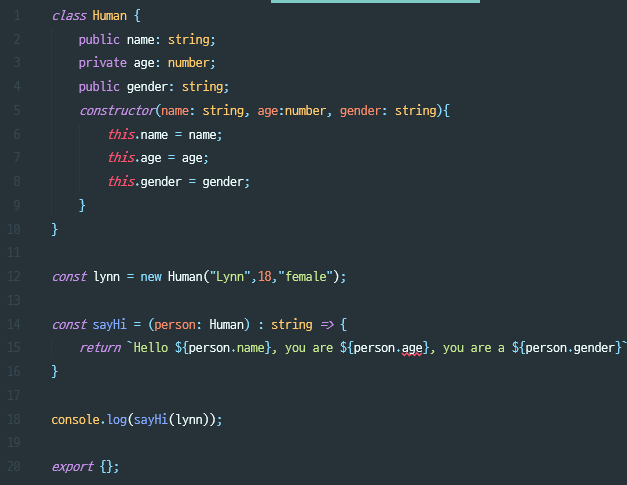
Class

interface를 쓸 때도 있고 class를 쓸 때도 있다. 타입스크립트 개발자는 이 두 가지를 자유자재로 쓸 줄 알아야 한다.

인터페이스가 아니라 class를 써야할 상황은 두 가지다.

1. js파일에 꼭 class부분을 나타내주고 싶을 때: 자바스크립트엔 interface 개념이 없기 때문에 컴파일된 js엔 안나와있다.

2. private, public 명시를 통해 클래스 내의 필드나 메소드를 보호하고 싶을 때.



이 경우엔 person.age 부분에서 에러가 난다. 왜냐면 클래스 바깥에서 필드를 호출했기 때문.