

** 주식 1 **

** 주식 2 **



ITEM 48. API주식10에 TSDOC사용하기

EDITED '23. 03. 11. (SAT) 신현호(@SWARVY)

* 본 사진은 개인의 사상과는 무관합니다



앞에 주석이 달려있고,
알기 쉬운 코드가 있어요.

이걸 더욱 보기 쉽게 할 수 있을까요?

```
// 인사말을 생성합니다. 결과는 보기 좋게 꾸며집니다.  
function greet(name: string, title: string) {  
  return `Hello ${title} ${name}`;  
}
```

```
/** 인사말을 생성합니다. 결과는 보기 좋게 꾸며집니다. */  
function greetJSDoc(name: string, title: string) {  
  return `Hello ${title} ${name}`;  
}
```



이럴 때는, JSDoc 스타일의 주석을 사용하세요

사용자를 위한 문서라면 더욱 효과적이에요!



JSDoc 스타일의 주석을 사용하면
함수에 붙어있는 주석을 툴팁에 표시해줘요

```
(alias) greetJSDoc(name: string, title: string): string  
import greetJSDoc
```

Generate a greeting. Result is formatted for display.

```
greetJSDoc('John Doe', 'Sir');
```

```
(alias) greet(name: string, title: string): string  
import greet
```

```
greet('John Doe', 'Sir');
```

TS 가 JSDoc 스타일을 지원하므로
적극적으로 사용하는게 좋아요

공개 API에 주석을 붙인다면
JSDoc 형태로 작성해야해요

```
/**
 * 인사말을 생성합니다.
 * @param name 인사할 사람의 이름
 * @param title 그 사람의 칭호
 * @returns 사람이 보기 좋은 형태의 인사말
 */
function greetFullTSDoc(name: string, title: string) {
  return `Hello ${title} ${name}`;
}
```

@param, @returns 같은
일반적 규칙도 적용 가능해요

```
/** 특정 시간과 장소에서 수행된 측정 */
interface Measurement {
  /** 어디에서 측정되었나? */
  position: Vector3D;
  /** 언제 측정되었나? epoch에서부터 초 단위로 */
  time: number;
  /** 측정된 운동량 */
  momentum: Vector3D;
}
```

당연히 타입에도 적용이 가능해요

타입 스크립트 관점에서는
TSDoc라고 부르기도 한답니다!

TSDoc 주석은 마크다운형식으로 작성해요

```
/**
 * 이 _interface_는 **세 가지** 속성을 가집니다.
 * 1. x
 * 2. y
 * 3. z
 */
interface Vector3D {
  x: number;
  y: number;
  z: number;
}
```

```
/**
 * This _interface_
 * 1. x
 * 2. y
 * 3. z
 */
```

```
export interface Vector3D {
  x: number;
  y: number;
  z: number;
}
```

interface Vector3D

This interface has **three** properties:

- 1. x
- 2. y
- 3. z

주의!!!

주석을 수필처럼 쓰지 마세요

주석에 타입의 정보를 명시하지 마세요