

Item 60. allowJS로 타임스크립트와 자바스크립트 같이 사용하기

'23. 4. 1. (SAT) 신현호(@SWARVY)



소규모 프로젝트는 한꺼번에 타입스크립트로 전환될 수 있지만
대규모 프로젝트는 한번에 전환이 안되므로 점진적으로 바꿔어야해요

근데, 이렇게 하려면 마이그레이션 기간 중에
자바스크립트와 타입스크립트가 동시에 작동할 수 있어야합니다!

1. tsconfig.json

- js 업로드 수용

```
{  
  "compilerOptions": {  
    ...  
    "allowJs": true,  
    //ts 프로젝트에서 js를 읽을지 여부, default: false  
    ...  
  },  
}
```

json

자바스크립트와 타입스크립트가 공존하기 위해서는
allowJs 컴파일러 옵션인데,
이걸 사용하면 TS파일과 JS파일을 서로 import 할 수 있습니다

allowJs를 설정하면 JS파일은 따로 건드릴 것이 없습니다.
@ts-check 지시자를 추가하지 않는다면요

타임 체크와는 관련이 없지만,
기존 빌드 과정에 타임스크립트 컴파일러를 추가하기 위해서도 allowJS 옵션이 필요해요

이외에도, 모듈 단위로 타임스크립트로 전화하는 과정에서 테스트를 수행해야하기에
이 또한 allowJS 옵션이 필요해요

```
module.exports = {  
  transform: {  
    '^.+\\.tsx?$': 'ts-jest',  
  },  
};
```

번들러에 타입스크립트가 통합되어 있거나, 플러그인 방식으로 통합이 가능하다면
allowJs를 간단히 적용할 수 있어요

```

/* 기본 옵션
 * -----
// "incremental": true,           /* 증분 컴파일 활성화 */
"target": "es5",                 /* ECMAScript 목표 버전 설정: 'ES3' (기본), 'ES5', '
"module": "esnext",             /* 생성될 모듈 코드 설정: 'none', 'commonjs', 'amd'
"lib": ["dom", "dom.iterable", "esnext"], /* 컴파일 과정에 사용될 라이브러리 파일 설정 */
"allowJs": true,                /* JavaScript 파일 컴파일 허용 */
// "checkJs": true,             /* .js 파일 오류 리포트 설정 */
"jsx": "react",                 /* 생성될 JSX 코드 설정: 'preserve', 'react-native
/* '.d.ts' 파일 생성 설정 */
// "declaration": true,         /* 해당하는 각 '.d.ts' 파일에 대한 소스 맵 생성 */
// "declarationMap": true,      /* 소스맵 '.map' 파일 생성 설정 */
// "sourceMap": true,           /* 복수 파일을 묶어 하나의 파일로 출력 설정 */
// "outFile": "./",             /* 출력될 디렉토리 설정 */
// "outDir": "./dist",          /* 입력 파일들의 루트 디렉토리 설정. --outDir 옵션을 사용
// "rootDir": "./",             /* 프로젝트 컴파일 활성화 */
// "composite": true,           /* 증분 컴파일 정보를 저장할 파일 지정 */
// "tsBuildInfoFile": "./",     /* 출력 시, 주석 제거 설정 */
// "removeComments": true,      /* 출력 방출(emit) 유무 설정 */
"noEmit": true,                 /* 'tslib'로부터 헬퍼를 호출할 지 설정 */
// "importHelpers": true,       /* 'ES5' 혹은 'ES3' 타겟 설정 시 Iterables 'for-of
// "downlevelIteration": true,  /* 각 파일을 별도 모듈로 변환 ('ts.transpileModule'과
"isolatedModules": true,

```

프레임워크 없이 빌드체인을 직접 구성한다면 복잡한 작업이 필요해요
한가지 방책으로는 outDir 옵션이 있습니다.

outDir 옵션을 사용하면 타입스크립트가 outDir에 지정된 디렉터리에
소스 디렉터리와 비슷한 구조로 JS 코드를 생성하게 되고,
outDir로 지정된 디렉터리를 대상으로 기존 빌드 체인을 실행하면 됩니다!