아이템 60. allowJs로 타입스크립트와 자바스크립트 같이 사용하기

allowJs

- 자바스크립트와 타입스크립트가 동시 동작.
- 기존 빌드 과정에 타입스크립트 컴파일러 추가.
- 모듈단위로 타입스크립트 전환을 위한 테스트 수행.

* allowJs : JavaScript 파일의 컴파일을 허용하는 옵션

allowJs

* 번들러 : 애플리케이션에 필요한 모든 파일들을 모듈 단위로 나누어 최소한의 파일 묶음(번들)으로 만들어 낸다.

outDir

* outDir : 출력될 디렉토리 설정.

outDir

- 점진적 마이그레이션을 위해서 자바스크립트와 타입스크립트를 동시적으로 사용해야 하며 이를 위해서 allowJs 컴파일러 옵션을 사용하자!
- 대규모 마이그레이션 작업에 앞서 테스트와 빌드 체인에 타입스크립트를 적용하자!

아이템 61. 의존성 관계에 따라 모듈 단위로 전환하기

- **lodash 라이브러리**

npm install --save-dev @type/lodash

- **외부 API**

아이템 61. 의존성 관계에 따라 모듈 단위로 전환하기

- 선언되지 않은 클래스 멤버
- 타입이 바뀌는 값

한꺼번에 객체 생성  객체 전개 연산자 . . . 사용

- JSDoc
- @ts-check

아이템 60 allowJs로 타입스크립트와 자바스크립트 같이 사용하기

- 마이그레이션 첫 단계는 서드파티 모듈과 외부 API 호출에 대한 @types를 추가하는 것이다.
- 의존성 관계도 : 아래에서 위로 올라가며 마이그레이션을 하면 된다.
  첫번째 모듈은 보통 유틸리티 모듈!
- 이상한 설계를 발견해도 리팩터링을 하지 말고 타입스크립트 전환에 집중 해야 한다.
- 타입스크립트로 전환하면서 발견되는 일반적인 오류들은 놓치지 말아야한다.
- 타입 정보 유지를 위해 필요에 따라서 JSDoc 주석을 활용할 수 있다.

아이템 62. 마이그레이션 완성을 위해 noImlicitAny 설정하기

noImlicitAny 설정

* noimplcitAny : any 타입으로 암시한 표현식과 선언에 오류를 발생시킨다.

아이템 62. 마이그레이션 완성을 위해 noImlicitAny 설정하기

- noImlicitAny 설정을 통해 마이그레이션 마지막 단계를 진행 해야 한다.
- noImlicitAny를 적용하기에 앞서 로컬에서만 설정하고 작업하며 타입 오류를 점진적으로 수정해야한다.
- 엄격한 타입체크 적용 전에 모든 팀원들이 타입스크립트에 익숙해져야 한다.