

46장. 제너레이터와 async/await

46.1 제너레이터란?

▼ 제너레이터란?

코드 블록의 실행을 일시 중지했다가 필요한 시점에 재개할 수 있는 특수한 함수

▼ 제너레이터와 일반 함수의 차이

1. 제너레이터 함수는 함수 호출자에게 함수 실행의 제어권을 양도할 수 있음
2. 제너레이터 함수는 함수 호출자와 함수의 상태를 주고 받을 수 있음
3. 제너레이터 함수를 호출하면 제너레이터 객체를 반환

46.2 제너레이터 함수의 정의

▼ 제너레이터 함수 선언법

function* 키워드로 선언 후, 하나 이상의 yield 표현식을 포함

▼ 애스터리스크(*) 위치

function 키워드와 함수 이름 사이라면 어디든지 상관없음

→ 하지만 일관성 유지를 위해 function 키워드 바로 뒤에 붙이는 것을 권장

46.3 제너레이터 객체

▼ 제너레이터 객체란?

이터러블이면서 동시에 이터레이터이다

46.4 제너레이터의 일시 중지와 재개

▼ yield 키워드 반환

제너레이터 함수의 실행을 일시 중지하거나 yield 키워드 뒤에 오는 표현식의 평가 결과를 제너레이터 함수 호출자에게 반환

46.5 제너레이터의 활용

- ▼ 제너레이터보다 간단하고 가독성 좋게 비동기 처리를 동기처리처럼 동작하는 방법
 `async/await`
- ▼ `async` 함수
 `async` 키워드를 사용해 정의하며 언제나 프로미스를 반환함
- ▼ `await` 함수
 `await` 키워드는 반드시 프로미스 앞에서 사용해야 함