Day 2 - Asynchronous programming (비동기 프로그래밍)

동기 vs 비동기

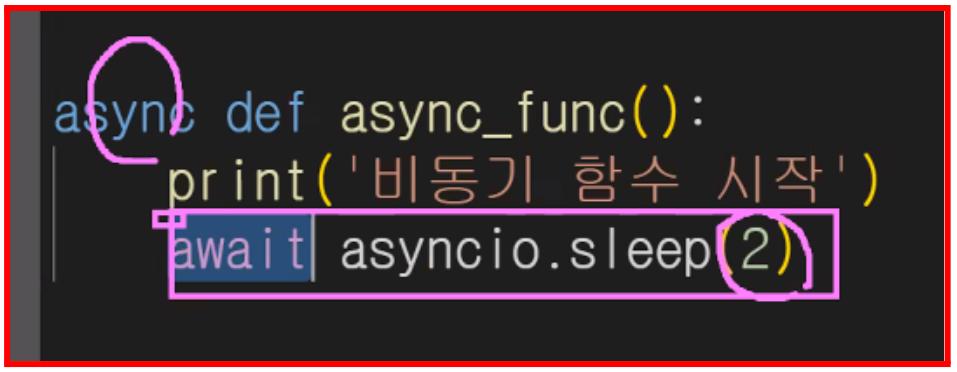
동기

- 중요한 정보가 전부 로딩되지 않은 상태에서 기능을 사용하는걸 막기 위해서 사용하는게 동기 (로딩창 띄우기 등)

비동기

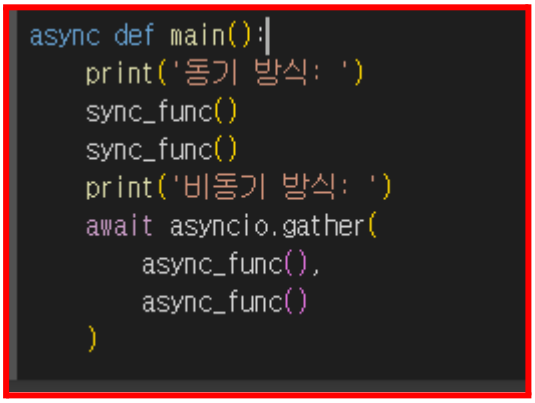
- 로딩이 전부 다 안되어도 상관없는것들 (사진 등)

비동기 함수:



- async, await은 같이 사용한다.

await asyncio.gather()



- 두개 이상을 한번에 여러개 실행시킬 때 사용한다.
- 동시에 같이 돈다

비동기 프로그래밍을 사용하는 이유

1. 응답성(Responsiveness)

- 비동기 작업은 여러 작업을 동시에 처리하고, 작업이 완료되기를 기다리는 동안 다른 작업을 처리할 수 있습니다. 이로써 프로그램의 응답성이 향상되고, 장기 실행 작업이 다른 작업을 차단하지 않도록 할 수 있습니다. 예를 들어, 네트워크 요청이나 데이터베이스 조회 등의 I/O 작업을 비동기로 처리하면, 다른 작업을 수행하면서 응답을 기다릴 필요가 없어집니다.

2. 확장성(Scalability)

- 비동기 프로그래밍은 많은 수의 동시 요청 또는 작업을 처리할 때 유용합니다. 여러 작업을 동시에 실행하고 완료될 때까지 기다리지 않아도 되므로, 처리량과 처리 속도를 향상시킬 수 있습니다.

3. 자원 효율성(Resource Efficiency)

- 비동기 작업은 작업의 대기 시간 동안 자원을 효율적으로 활용할 수 있습니다. 대기 시간 동안 CPU나 메모리 등의 자원을 다른 작업에 할당하여 더 많은 작업을 처리할 수 있습니다.

4. 병렬성(Concurrency)

- 비동기 프로그래밍은 병렬적인 작업을 효율적으로 처리할 수 있는 방법을 제공합니다. 여러 작업이 동시에 실행될 수 있으며, 이는 멀티코어 프로세서를 활용하여 작업을 분산 처리할 수 있음을 의미합니다.

웹사이트:
https://www.tcpschool.com/json/intro
https://www.w3schools.com/

JSON

- JSON은 JavaScript Object Notation의 약자입니다.
- JSON은 좀 더 쉽게 데이터를 교환하고 저장하기 위하여 만들어진 텍스트 기반의 데이터 교환 표준입니다.
- **기종, 어플, 웹 상관없이 데이터를 쉽게 주고받기위해 만들어졌다.**

Json 특징:

1. JSON은 자바스크립트를 확장하여 만들어졌습니다.
2. JSON은 자바스크립트 객체 표기법을 따릅니다.
3. JSON은 사람과 기계가 모두 읽기 편하도록 고안되었습니다.

JSON 문법:

- JSON은 자바스크립트의 객체 표기법에서 리터럴(literal)과 프로퍼티(property)를 표현하는 방법만 가져와서 사용합니다.
- 따라서 JSON 데이터는 모양과 규칙이 매우 단순합니다.
- 그로 인해 브라우저 영역에서도 쉽고 빠르게 그 의미를 해석할 수 있으며, 다른 프로그래밍 언어에서도 구현하기 쉽습니다.

예제

```
{
    "name": "식빵",
    "family": "웰시코기",
    "age": 1,
    "weight": 2.14
}
```

- “Key” : value