정의

1. 블록과 논블록

* 블록 : 자신의 수행결과가 끝날 때 까지 제어권을 갖고 있는 것
  + A라는 함수를 호출했을 때, A를 호출했을 때의 기대행위를 모두 끝마칠때까지 기다렸다가 리턴
  + 리턴값에 대한 신뢰성이 있다.
* 논블록 : 자신이 호출되었을 때 제어권을 바로 자신을 호출한 쪽으로 넘기며, 자신을 호출한 쪽에서 다른 일을 할 수 있도록 하는 것
  + A라는 함수를 호출했을 때, A를 호출했을 때의 기대행위를 요청하고 바로 리턴
  + 리턴값에 대한 신뢰성을 보장하지 못한다.

1. 동기와 비동기

* 동기 : 요청과 그 결과가 동시에 일어남
  + 어떤 객체 또는 함수 내부에서 다른 함수를 호출했을 때 이 함수의 결과를 호출한 쪽에서 처리
  + A라는 행위와 B라는 행위가 존재하고 A와 B가 순차적으로 작동하거나, A가 별개의 행위가 아닌 B를 관찰하는 행위라면 이것이 동시에 작동하더라도 동기이다.
* 비동기 : 요청과 그 결과가 동시에 일어나지 않음
  + 어떤 객체 또는 함수 내부에서 다른 함수를 호출했을 때 이 함수의 결과를 호출한 쪽에서 처리하지 않고 호출된 함수 혼자 처리
  + A라는 행위와 B라는 행위가 존재하고 A와 B가 동시에, 또는 순차적이지 않게 작동하면 비동기이다. 이 때, A와 B 사이에는 인과관계가 존재해야 하며 인과관계가 존재하지 않으면 비동기라고 볼 수 없다.

각각의 조합

1. 블록 / 동기
   * 호출된 함수가 자신이 할 일을 모두 마칠 때까지 제어권을 계속 가지고(블록) 수행 결과 및 종료를 호출한 함수 쪽에서 처리(동기)
   * 작업을 수행중일 때 다른 작업을 병행할 수 없음
2. 블록 / 비동기
   * 호출된 함수가 자신이 할 일을 모두 마칠 때까지 제어권을 계속 가지고(블록) 수행 결과 및 종료를 호출된 함수가 신경쓰고 처리함(비동기)
   * 작업을 수행중일 때 다른 작업을 병행할 수 있으나 데이터를 대기하면서 유휴상태로 있기 때문에 프로그램의 리소스가 낭비되는 단점이 있다.
3. 논블록 / 동기
   * 호출된 함수가 제어권을 바로 호출한 함수 쪽으로 넘기고 다른 일을 할 수 있도록 하고(논블록) 수행 결과 및 종료를 호출한 함수 쪽에서 처리(동기)
4. 논블록 / 비동기
   * 호출된 함수가 제어권을 바로 호출한 함수 쪽으로 넘기고 다른 일을 할 수 있도록 하고(논블록) 수행 결과 및 종료를 호출된 함수가 신경쓰고 처리함(비동기)
   * 하나의 스레드로 많은 작업을 수행할 수 있다. 현재의 자원에서 더 높은 효율성을 추가하는 방법