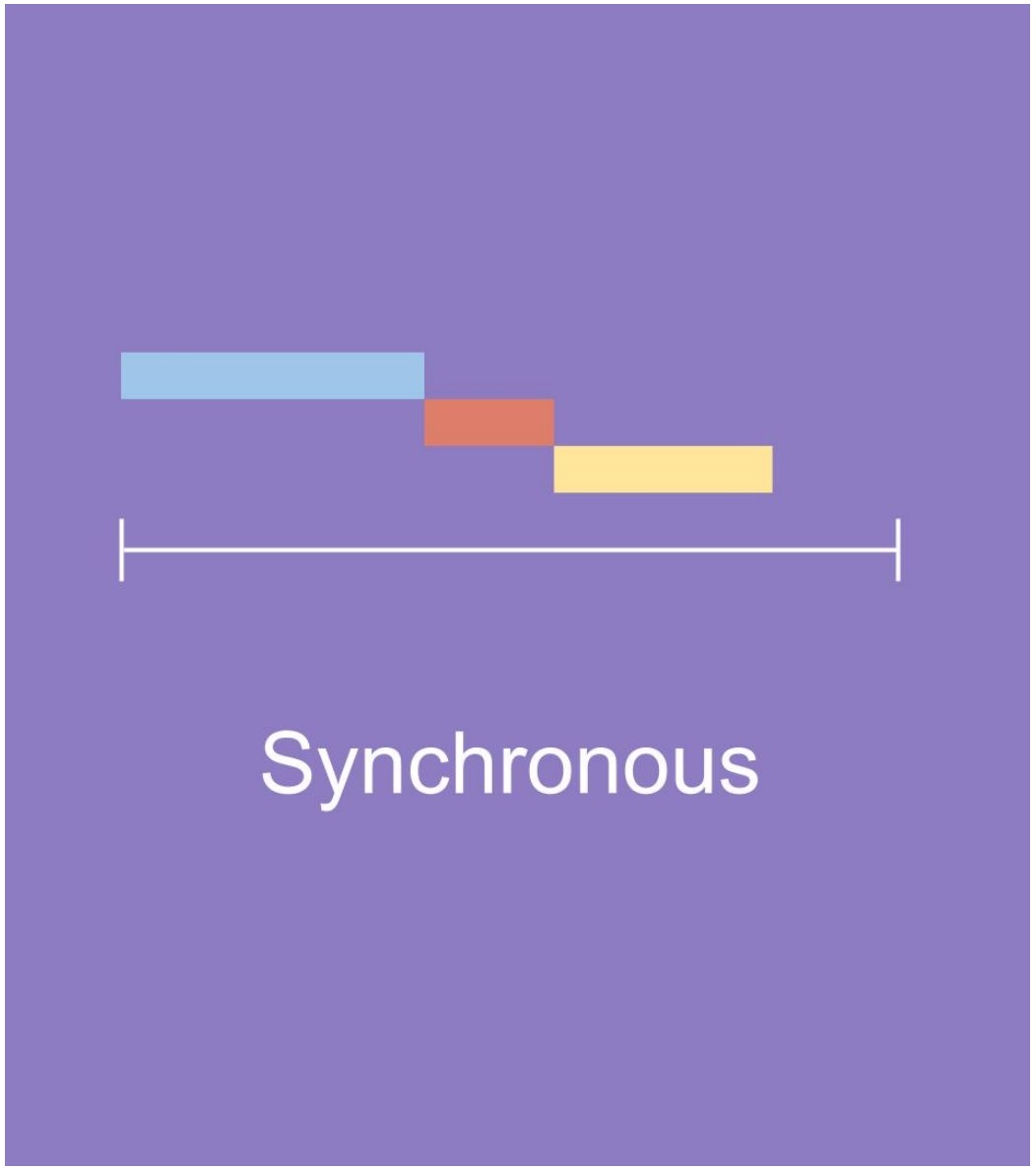
동기와 비동기

이번에는 이해해보자!

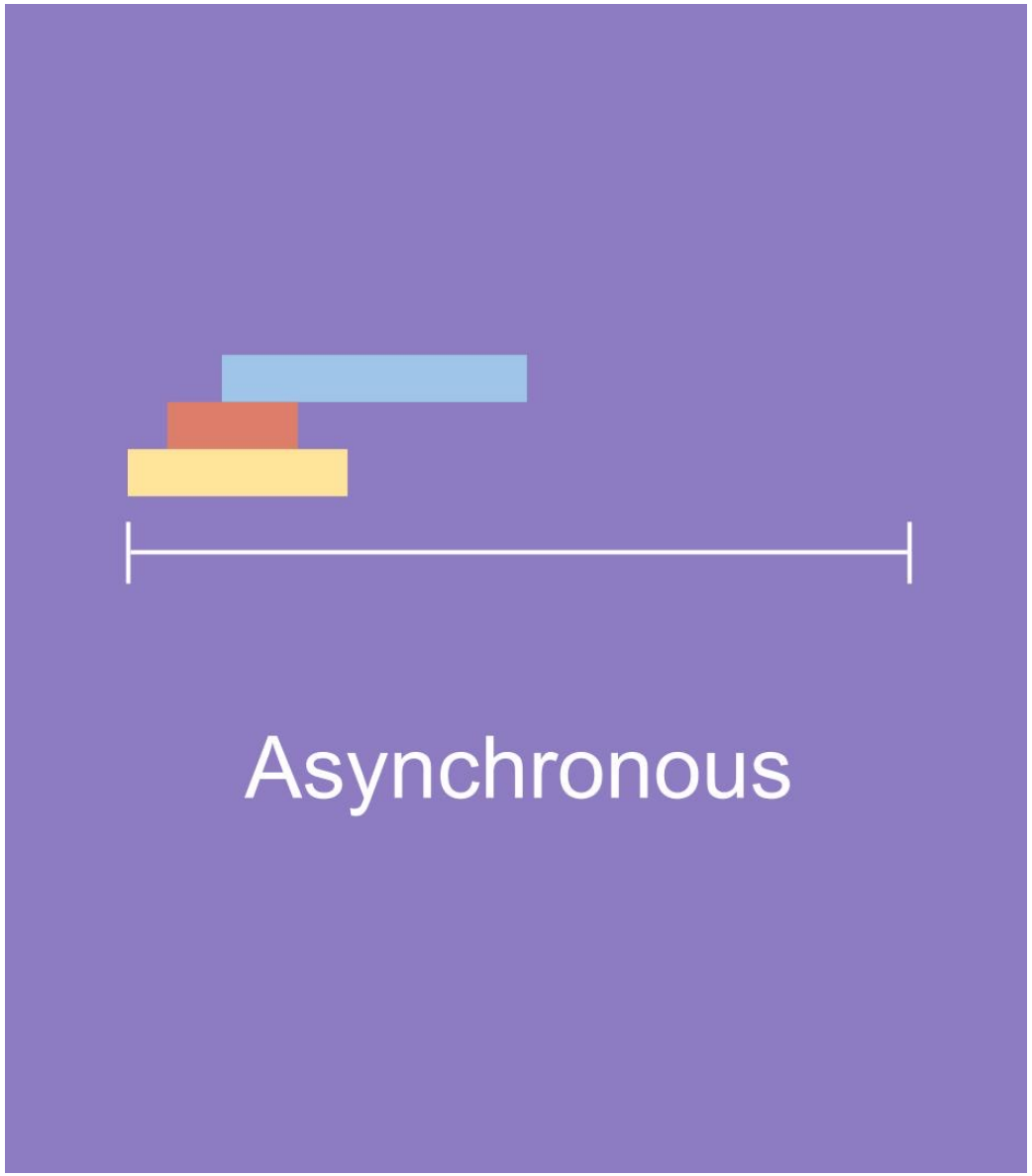
동기 (Synchronous)

요청과 결과가 동시에 일어나는 약속
요청을 하면 결과가 주어져야함
호출한 함수가 작업 완료 여부를 확인



비동기 (Asynchronous)

요청시 결과를 바로 주어주지 않음
-> 먼저 수행한 함수가 있어도
그 함수의 결과를 받지 않아도
다른 함수를 수행할 수 있음
Callback으로 자신의 작업 완료 여부를 알려줌



블로킹(Blocking) / 논 블로킹(Nonblocking)

제어권의 문제
- 블로킹 : 제어권을 넘겨주지 않음
- 논 블로킹 : 제어권을 넘겨줌

함수 A안에 B를 호출 했을 경우를 가정

함수 A가 B를 호출함 (제어권이 함수 B에게 주어짐)

- 블로킹 : 함수 B가 할 일을 다 마칠 때까지 제어권을 가지고 있음
- 논 블로킹 : 함수B가 할 일을 다 마치지 않아도 A에게 제어권을 넘겨줌 -> A는 B가 진행중에 다른 일을 할 수 있음

논블로킹과 비동기가 같은 거 아니야?? 다른 일을 할 수 있잖아!!

아님!!! 비슷한 거 같지만 맥락이 다름!!!!

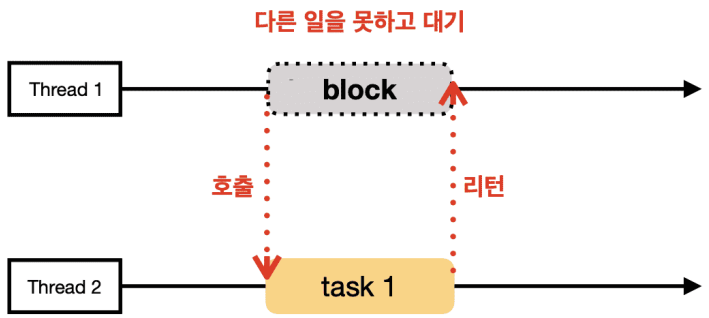
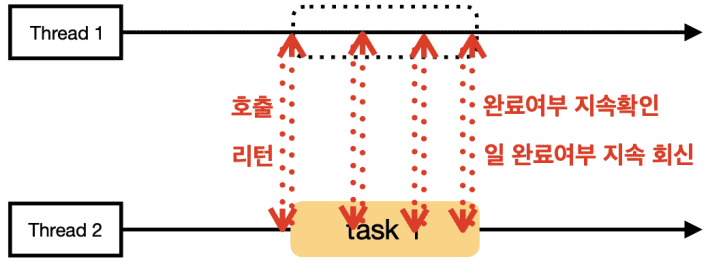
비동기는 나중에 *결과값을 보내줍니다.
하지만!! 논블로킹은 결과값이 있든 없든 보내줍니다!!(없으면 에러값)

논블로킹은 소켓으로 생각하시면 편합니다!

A : 나 C 할려고
B : 안됨
A : oo
…
A : 나 C 할려고 (polling)
B : 안됨
A : oo
…
A : 나 C 할려고
B : ㅇㅇ됨
A : 되었다!

바로 리턴하지 않음

바로 리턴

	Blocking (호출된 함수 여기서는 task1이 제어권을 가짐)	Non - Blocking (호출한 함수가 제어권을 가짐)
호출한 함수 동기 (sync) (호출한 함수가 작업 완료 여부를 확인)		
Callback 함수 비동기 (async) (Callback함수가 작업 완료 여부를 확인)