

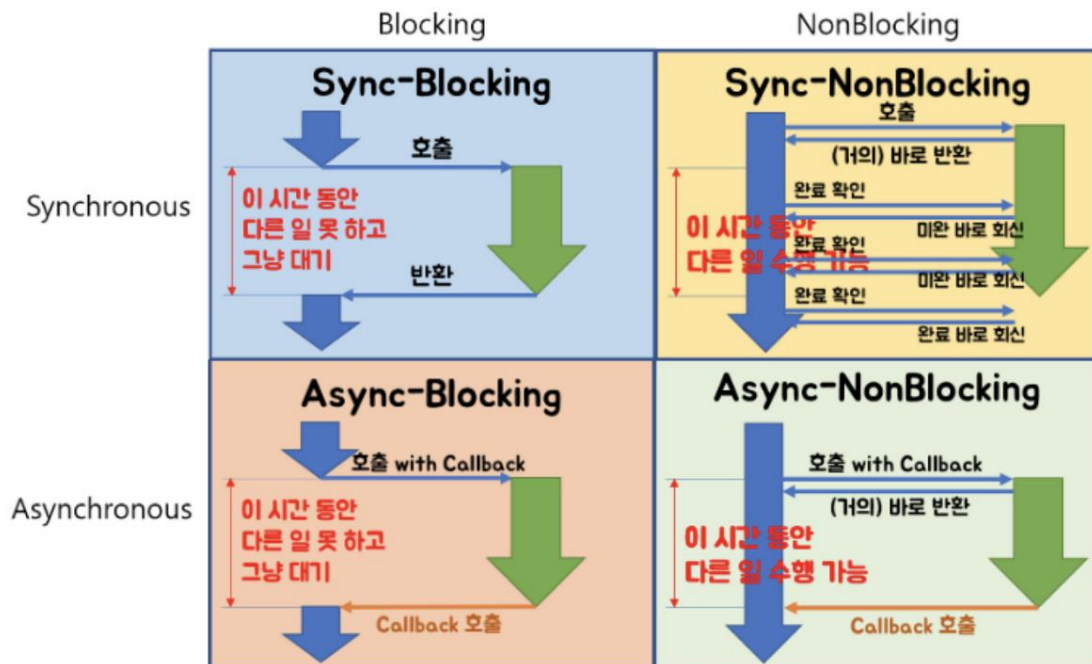
<Blocking/Non-blocking & Synchronous/Asynchronous>

동기/비동기는 우리가 일상 생활에서 많이 들을 수 있는 말이다.

Blocking과 Synchronous, 그리고 Non-blocking과 Asynchronous를

서로 같은 개념이라고 착각하기 쉽다.

각자 어떤 의미를 가지는지 간단하게 살펴보자



1. Blocking/Non-blocking

블럭/논블럭은 간단히 말해서 호출된 함수가 호출한 함수에게 제어권을 건네주는 유무의 차이라고 볼 수 있다.

함수 A, B가 있고, A 안에서 B를 호출했다고 가정해보자. 이때 호출한 함수는 A고, 호출된 함수는 B가 된다. 현재 B가 호출되면서 B는 자신의 일을 진행해야 한다. (제어권이 B

에게 주어진 상황)

Blocking : 함수 B는 내 할 일을 다 마칠 때까지 제어권을 가지고 있다. A는 B가 다 마칠 때까지 기다려야 한다.

Non-blocking : 함수 B는 할 일을 마치지 않았어도 A에게 제어권을 바로 넘겨준다. A는 B를 기다리면서도 다른 일을 진행할 수 있다.

즉, 호출된 함수에서 일을 시작할 때 바로 제어권을 리턴해주느냐, 할 일을 마치고 리턴해주느냐에 따라 블럭과 논블럭으로 나누어진다고 볼 수 있다.

2. Synchronous/Asynchronous

동기/비동기는 일을 수행 중인 동시성에 주목하자

1번에서 처럼 함수 A와 B라고 똑같이 생각했을 때, B의 수행 결과나 종료 상태를 A가 신경 쓰고 있는 유무의 차이라고 생각하면 된다.

Synchronous : 함수 A는 함수 B가 일을 하는 중에 기다리면서, 현재 상태가 어떤지 계속 체크한다.

Asynchronous : 함수 B의 수행 상태를 B 혼자 처리한다. (Callback)

즉, 호출된 함수(B)를 호출한 함수(A)가 신경쓰는지, 호출된 함수(B) 스스로 신경쓰는지를 동기/비동기라고 생각하면 된다.

비동기는 호출시 Callback을 전달하여 작업의 완료 여부를 호출한 함수에게 답하게 된다. (Callback이 오기 전까지 호출한 함수는 신경쓰지 않고 다른 일을 할 수 있음)

위 그림처럼 총 4가지의 경우가 나올 수 있다. 이걸 좀 더 이해하기 쉽게 Case 별로 예를 통해 보면서 이해하고 넘어가보자

상황 : 치킨집에 직접 치킨을 사러감

1) Blocking & Synchronous

나 : 사장님 치킨 한마리만 포장해주세요

사장님 : 네 금방되니까 잠시만요!

나 : 냇

-- 사장님 치킨 튀기는 중--

나 : (아 언제 되지?...궁금한데 그냥 멍뚱히 서서 치킨 튀기는거 보면서 기다림)

2) Blocking & Asynchronous

나 : 사장님 치킨 한마리만 포장해주세요

사장님 : 네 금방되니까 잠시만요!

나 : 앓 냇

-- 사장님 치킨 튀기는 중--

나 : (언제 되는지 안 궁금함, 잠시만이래서 다 될때까지 서서 붙잡힌 상황)

3) Non-blocking & Synchronous

나 : 사장님 치킨 한마리만 포장해주세요

사장님 : 네~ 주문 밀려서 시간 좀 걸리니까 볼일 보시다 오세요

나 : 냇

-- 사장님 치킨 튀기는 중--

(5분뒤) 나 : 제꺼 나왔나요?

사장님 : 아직이요

(10분뒤) 나 : 제꺼 나왔나요?

사장님 : 아직이요ㅠ

(15분뒤) 나 : 제꺼 나왔나요?

사장님 : 아직이요ㅠㅠ

4) Non-blocking & Asynchronous

나 : 사장님 치킨 한마리만 포장해주세요

사장님 : 네~ 주문 밀려서 시간 좀 걸리니까 볼일 보시다 오세요

나 : 냇

-- 사장님 치킨 튀기는 중--

나 : (앉아서 다른 일 하는 중)

...

사장님 : 치킨 나왔습니다

나 : 잘먹겠습니다~

[Reference]

1. <http://homoefficio.github.io/2017/02/19/Blocking-NonBlocking-Synchronous-Asynchronous/>
2. <https://musma.github.io/2019/04/17/blocking-and-synchronous.html>