

blocking/non-blocking & synchronous/Asynchronous

Blocking & non-blocking

Blocking : 자신의 작업을 진행하다가 다른 주체의 작업이 시작되면 다른 작업이 끝날 때까지 기다렸다가 자신의 작업을 시작하는 것

- 특징
1. 요청한 작업을 마칠 때까지 계속 대기한다.
 2. 즉시 return한다.
 3. return 값을 받아야 끝난다.
 4. Thread 관점으로 본다면, 요청한 작업을 마칠 때까지 계속 대기하며 return 값을 받을 때까지 한 Thread를 계속 사용/대기 한다.

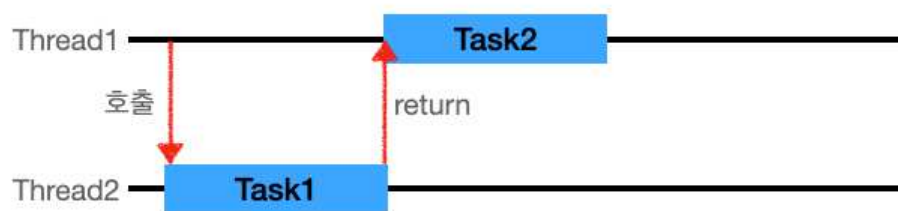
Non-Blocking : 다른 주체의 작업에 관련 없이 자신의 작업을 하는 것

- 특징
1. 요청한 작업을 즉시 마칠 수 없다면 즉시 return한다.
 2. 즉시 리턴하지 않는다. (일을 못하게 막는다.)
 3. Thread 관점으로 본다면, 하나의 Thread가 여러 개의 IO를 처리 가능하다.

즉, 다른 주체가 작업을 할 때, 자신의 제어권이 있는지 없는지의 차이

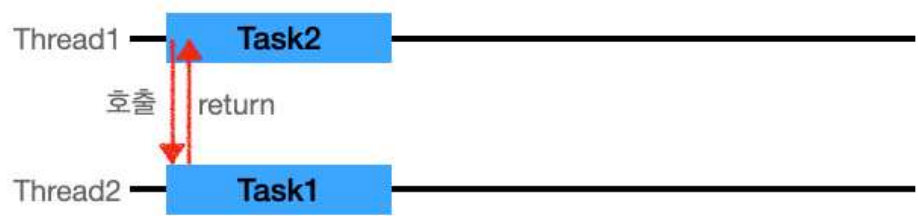
Synchronous & Asynchronous

Synchronous : 작업을 동시에 수행하거나, 동시에 끝나거나, '끝나는 동시에 시작함'을 의미



- 특징
1. Thread1이 작업을 시작 시키고, Task1이 끝날때까지 기다렸다가 Task2를 시작한다.
 2. 작업 요청을 했을 때 요청의 결과값(return)을 직접 받는 것이다.
 3. 요청의 결과값이 return값과 동일하다.
 4. 호출한 함수가 작업 완료를 신경 쓴다.

Asynchronous : 시작, 종료가 일치하지 않으며, 끝나는 동시에 시작을 하지 않음을 의미



- 특징
1. Thread1이 작업을 시작 시키고, 완료를 기다리지 않고, Thread1은 다른 일을 처리할 수 있다.
 2. 작업 요청을 했을 때 요청의 결과값(return)을 간접적으로 받는 것이다.
 3. 요청의 결과값이 return값과 다를 수 있다.
 4. 해당 요청 작업은 별도의 스레드에서 실행하게 된다.
 5. 콜백을 통한 처리가 비동기 처리라고 할 수 있다.
 6. 호출된 함수(callback 함수)가 작업 완료를 신경 쓴다.

즉, 결과를 돌려주었을 때 순서와 결과에 관련이 있는지 아닌지로 판단 가능

조합 Blocking/Sync & Non-Blocking/Sync & Blocking/Async & Non-Blocking/Async

	Blocking	Non-blocking
Synchronous	Read/Write	Read/Write (Polling)
Asynchronous	I/O Multiplexing (Select / Poll)	Asynchronous I/O

Blocking Sync : Blocking: 서류를 읽을 때까지 기다리세요.

이렇게 하세요! Sync :(바로처리)

ex) 자바 입력 요청 : 제어권이 넘어가기 때문에 입력 전까지 아무일도 안함,
입력 후 바로 처리

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        System.out.print("출력하고 싶은 메시지를 입력해주세요 : ");
        final Scanner scanner = new Scanner(System.in);
        String message = scanner.nextLine();
        System.out.println("블로킹 동기");
        System.out.println(message);
    }
}
```

Non-Blocking Sync : Non-Blocking: 서류를 읽어 볼테니 돌아가세요(중간중간 물어봄)

이렇게 하세요! Sync : (바로처리)

ex) 게임상에서 맵을 요청할 때

Blocking Async : Blocking : 서류를 읽을 때까지 기다리세요.

이렇게 하세요! Async : (처리를 안할수도 할수도?)

ex) 주로 개발자의 실수로 이렇게 되는 경우도 있음

Non-Blocking Async : Non-Blocking : 서류를 읽어 볼테니 돌아가세요(중간중간 물어봄)

이렇게 하세요! Async : (처리를 안할수도 할수도?)

ex) js 에서 api를 통해서 다른 작업을 하다가 추가적인 작업 처리를 할 때, 사용

```
fetch('url', option)
  .then((response) => {
    return response.json();
  })
  .then((data) => {
    something(data);
  });
```