

01

◦1-1. Overview

Coroutine

## 코루틴이란

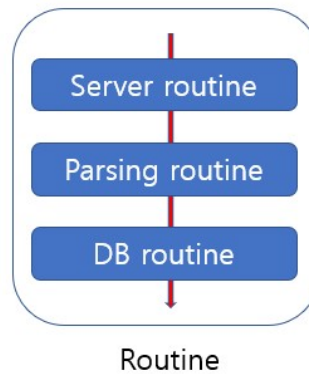
---

- 코루틴(Coroutine)은 코틀린 에서 제공하는 비동기 프로그램을 위한 기법 혹은 라이브러리
- C++, Go, 파이썬, 코틀린등이 코루틴을 지원하는 대표적인 언어

# Co-Routine

---

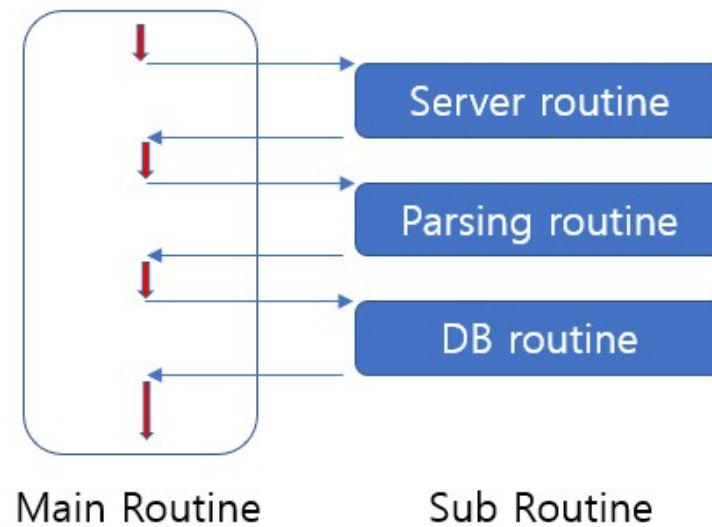
- 코루틴(Coroutine)의 이름에서 Co 는 with 혹은 together
- 루틴이란 소프트웨어 개발에서 흔히 사용되는 용어로 하나의 업무처리 단위



## sub-routine vs co-routine

---

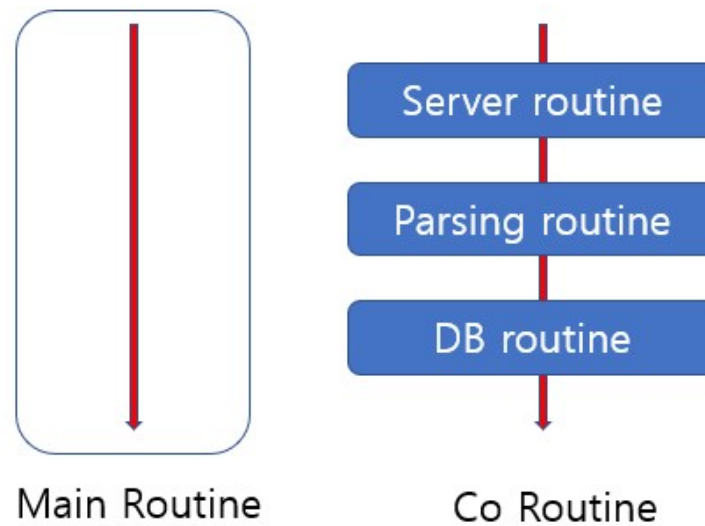
- sub-routine 은 단일 실행흐름에 의해 각 루틴들이 순차적으로 실행되는 구조
- 동기(Synchronous) 프로그래밍



## sub-routine vs co-routine

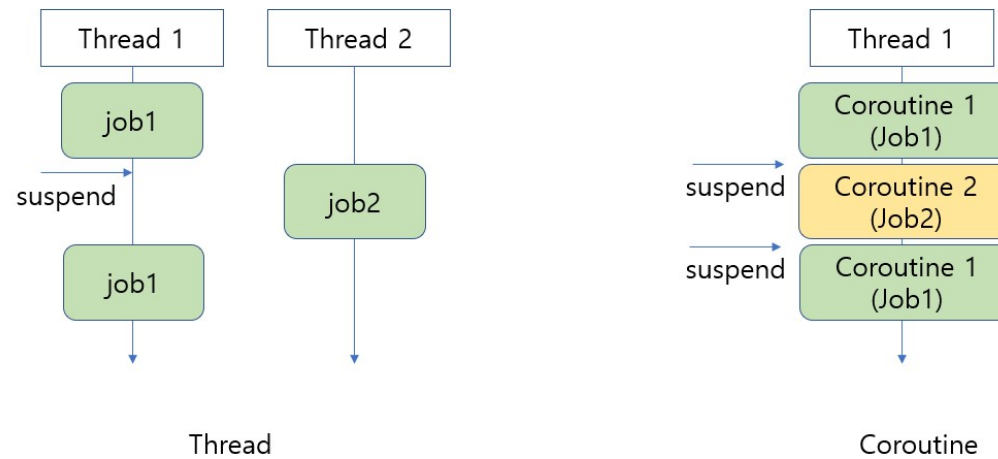
---

- Co – Routine 의 개념은 동시에 같이 실행되는 루틴을 의미
- 비동기(Asynchronous) 프로그래밍

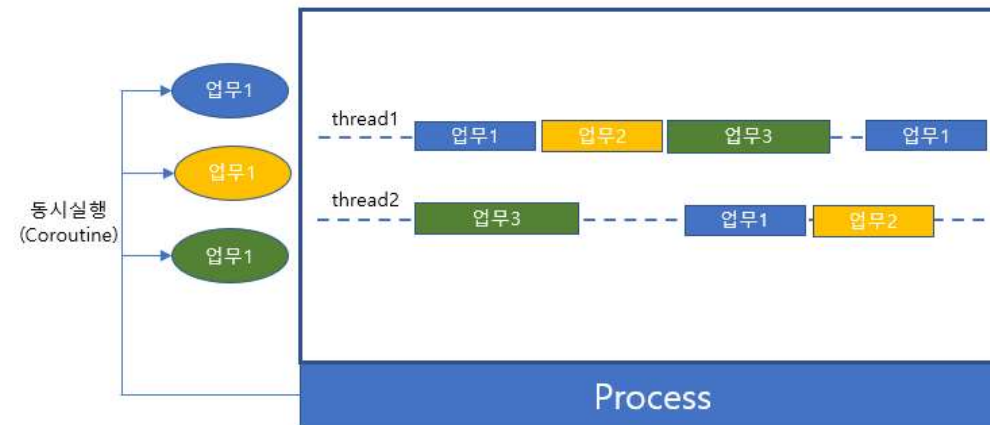
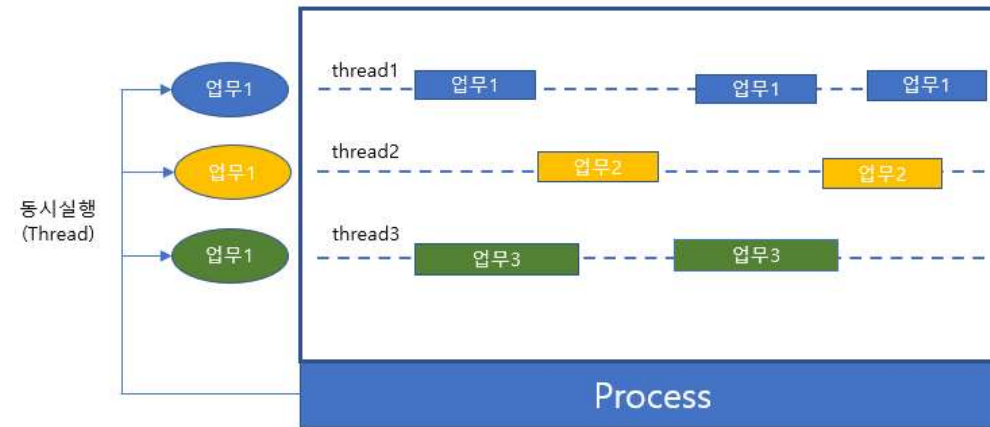


# Non-Blocking Lightweight Thread

---



# Non-Blocking Lightweight Thread



# 라이브러리 등록

---

- Gradle

```
dependencies {  
    implementation("org.jetbrains.kotlin:kotlinx-coroutines-core:1.7.3")  
}
```

- Maven

```
<dependency>  
  <groupId>org.jetbrains.kotlin</groupId>  
  <artifactId>kotlinx-coroutines-core</artifactId>  
  <version>1.7.3</version>  
</dependency>
```



## 코루틴 핵심 개념 - 빌더

---

- 코루틴을 만드는 역할
- launch, async, produce, actor

## 코루틴 핵심 개념 - Suspend Function

---

- suspend 예약어로 선언된 함수
- 중단 함수는 다른 중단 함수내에서만 호출 가능

**suspend fun** delay(timeMillis: Long)

- 함수를 실행시키는 스레드가 release 될 수 있는 함수

## 코루틴 핵심 개념 - Job

---

- 코루틴을 만들면 Job 객체 리턴
- 코루틴을 지칭하는 클래스
- Job 을 이용해 코루틴을 취소, 코루틴이 종료될 때까지 대기하는 등의 다양한 작업



# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare