13강

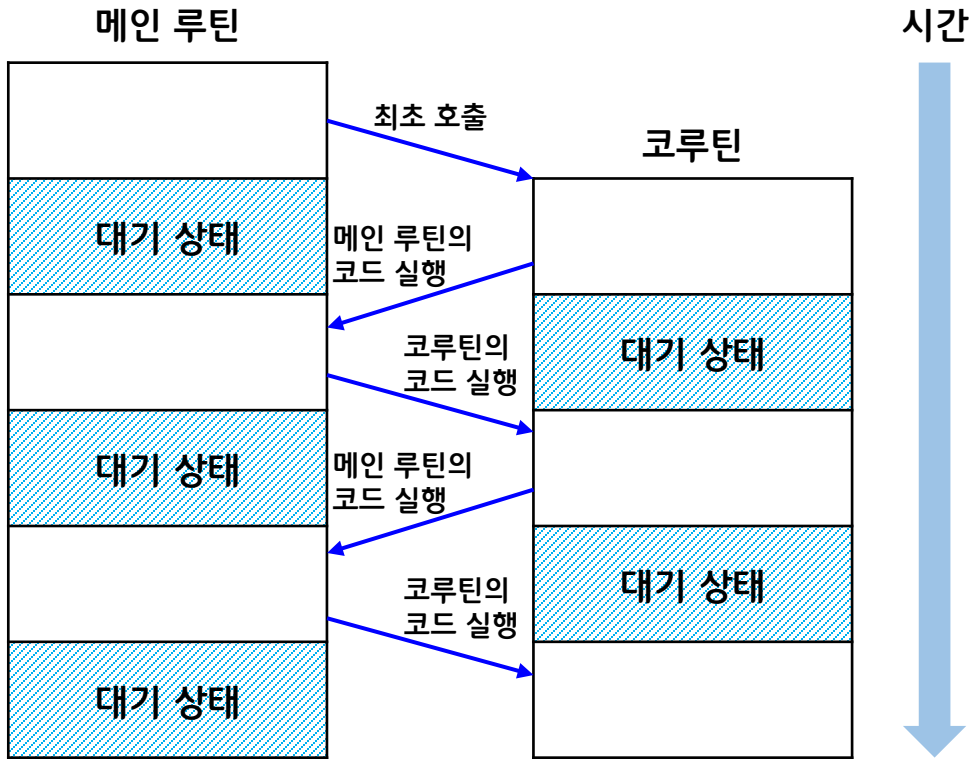
코루틴

13-1 코루틴 개요

- 코루틴
  - ✓ 일반적인 코드에서 메인 루틴으로부터 서브 루틴을 호출하면 서브 루틴의 코드를 실행한 뒤 다시 메인 루틴으로 돌아온다. 또한 서브 루틴이 끝나면 서브 루틴의 내용은 메모리에서 삭제된다.
  - ✓ 코루틴(coroutine)은 cooperative routine를 의미하는데 서로 협력하는 루틴 관계를 뜻한다.
  - ✓ 메인 루틴과 서브 루틴처럼 종속된 관계가 아니라 서로 대등한 관계이며 특정 시점에 상대방의 코드를 서로 실행할 수 있다.

13-1 코루틴 개요

- 코루틴
  - ✓ 코루틴은 함수가 종료되지 않은 상태에서 메인 루틴의 코드를 실행한 뒤 다시 돌아와서 코루틴의 코드를 실행한다.
  - ✓ 코루틴은 종료되지 않았으므로 코루틴의 내용도 메모리에 계속 유지된다.
  - ✓ 일반 함수를 호출하면 코드를 한 번만 실행할 수 있지만, 코루틴은 코드를 여러 번 실행할 수 있다.
  - ✓ 참고로 함수의 코드를 실행하는 지점을 진입점(entry point)이라고 하는데, 코루틴은 진입점이 여러 개인 함수라고 말할 수 있다.



13-2 코루틴 생성

- 코루틴 값 전송
  - ✓ 코루틴은 제너레이터의 특별한 형태로 제너레이터는 yield로 값을 발생시켰지만 코루틴은 yield로 외부 값을 받아올 수 있다.
  - ✓ 코루틴에 값을 전달하려면 send 메서드를 사용하면 된다.
  - ✓ Send 메서드가 전달한 값을 받으려면 yield를 괄호로 묶어준 뒤, 변수에 저장하여 사용하면 된다.

13-2 코루틴 생성

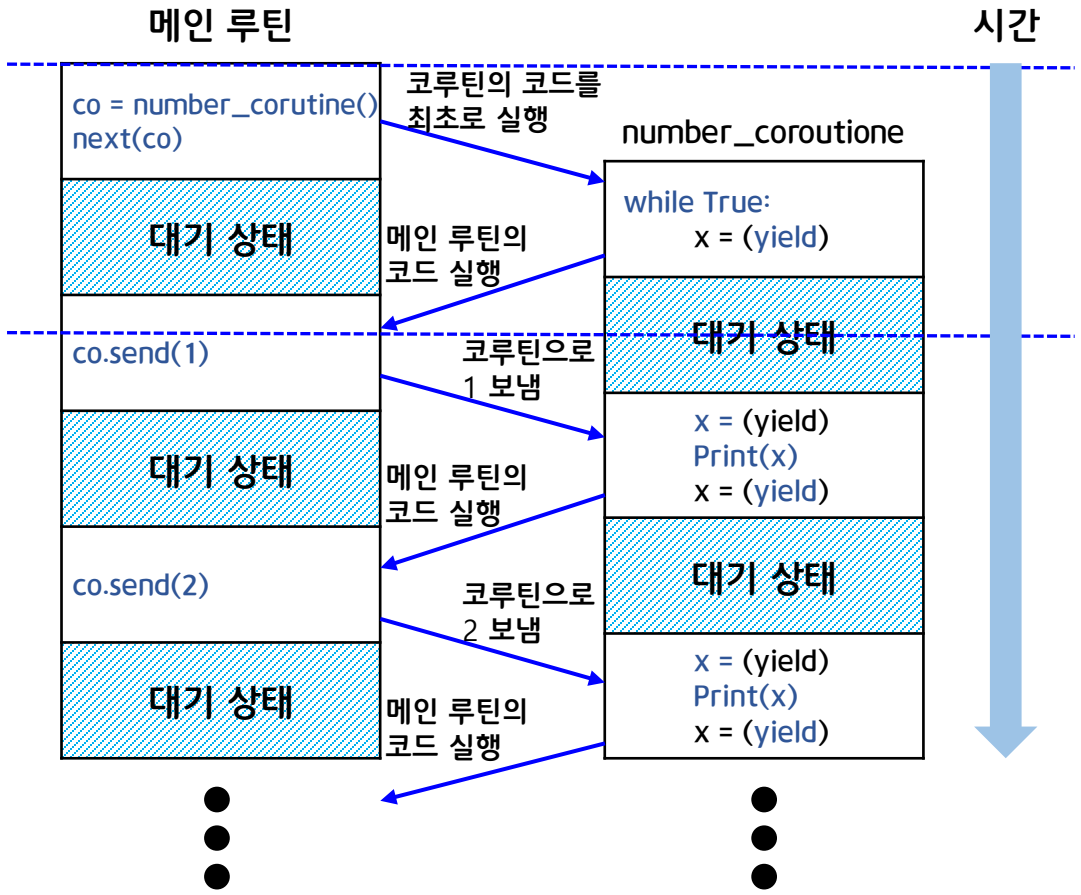
● 코루틴 값 전송

```
def number_corutine():
    while True:
        x = (yield)
        print(x)

co = number_corutine()
next(co)

co.send(1)
co.send(2)
co.send(3)
co.close()
```

```
[결과]
1
2
3
```



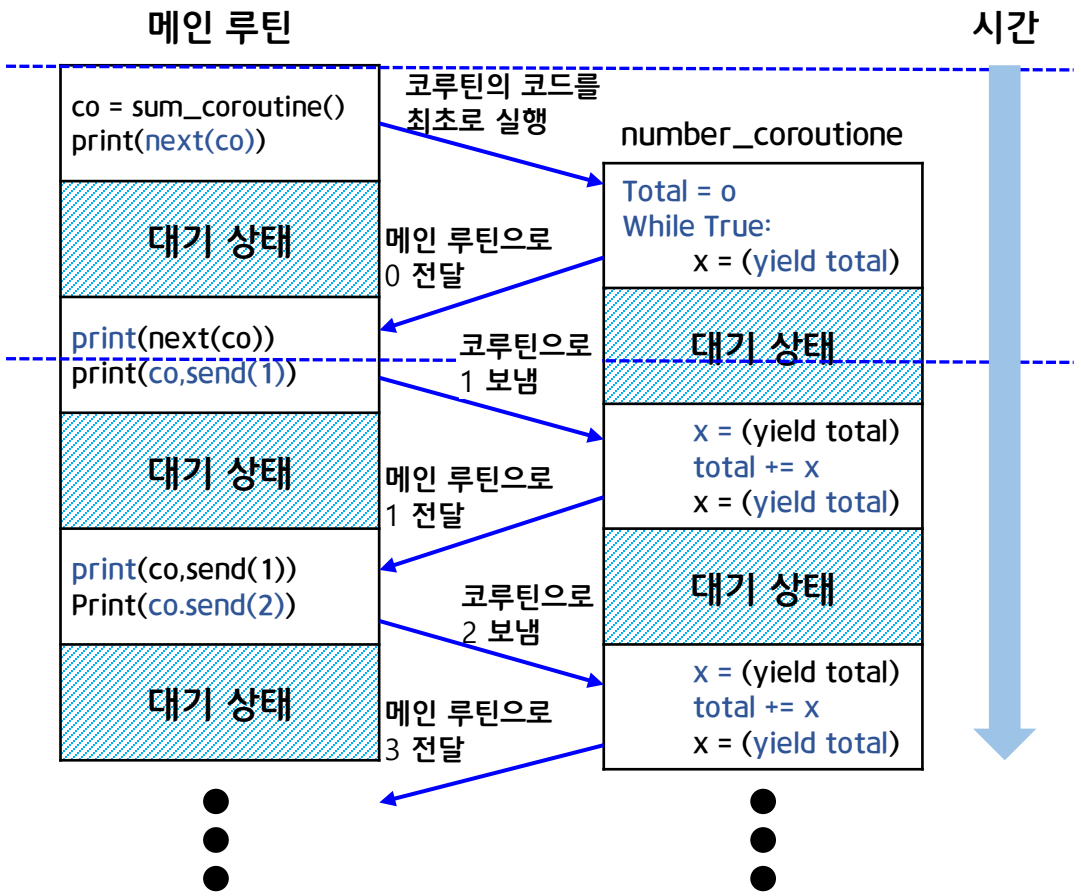
13-2 코루틴 생성

- 코루틴 리턴
  - ✓ (yield 변수) 형식으로 yield에 변수를 지정한 뒤 괄호로 묶어주면 값을 받아오면서 바깥으로 값을 리턴하면 된다.

```
def number_corutine2():
    total = 0
    while True:
        x = yield total
        total+=x

co = number_corutine2()
print(next(co))
print(co.send(1))
print(co.send(2))
print(co.send(3))
```

```
[결과]
0
1
3
6
```



13-3 코루틴 예외 처리

- 코루틴 예외 처리
  - ✓ 코루틴을 강제로 종료하고 싶은 면 close 메서드를 사용면 된다.
  - ✓ RuntimeError 예외를 발생하면 에러 메시지를 출력하고 누적된 값을 코루틴 바깥으로 전달한다.

```
def number_corutine3():
    try:
        while True:
            x = (yield)
            print(x)
    except GeneratorExit:
        print()
        print('corutine end!!!')

co = number_corutine3()
next(co)

for i in range(5):
    co.send(i)
co.close()
```

```
[결과]
1
2
3
4
```

```
def number_corutine4():
    total = 0
    try:
        while True:
            x = (yield)
            total += x
    except RuntimeError as e:
        print(e)
        yield  total

co = number_corutine4()
next(co)

for i in range(20):
    co.send(i)
print(co.throw(RuntimeError, '강제 예외 발생 코루틴 종료'))
```

```
[결과]
강제 예외 발생 코루틴 종료
190
```

13-3 코루틴 예외 처리

- 하위 코루틴
  - ✓ 코루틴에서 값을 yield from을 이용하여 전달 받을 수 있다.

```
def accumulate():
    total = 0
    while True:
        x = (yield)
        if x is None:
            return total
        total += x

def sum_corutine():
    while True:
        total = yield from accumulate()
        print(total)

co = sum_corutine()
next(co)

for i in range(20):
    co.send(i)

co.send(None)

for i in range(1,101):
    co.send(i)
co.send(None)
co.close()
```

```
[결과]
190
5050
```