

Chapter 05_003. 함수 내에서 또 다른 함수 호출

—

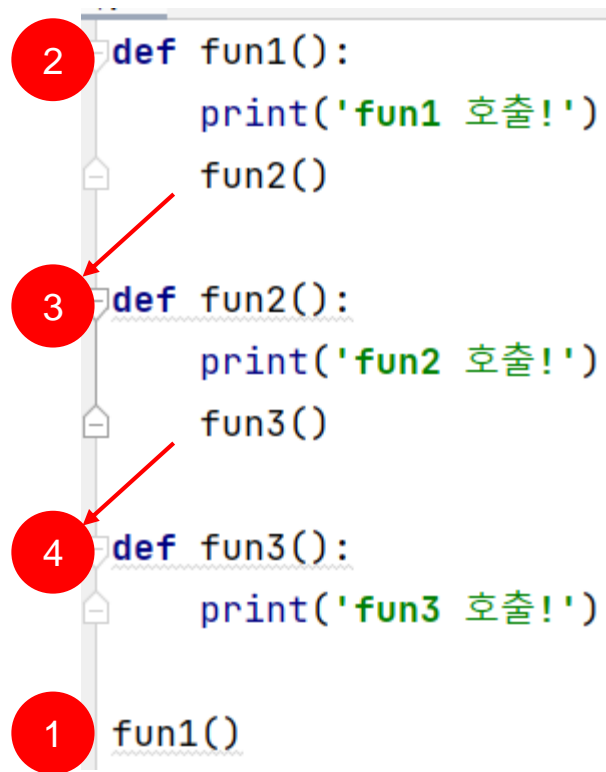
또 다른 함수를 호출하자!

또 다른 함수 호출

pass 사용

• 또 다른 함수 호출

➤ 함수 내에서 또 다른 함수를 호출할 수 있다.



```
2 def fun1():  
    print('fun1 호출!')  
    fun2()  
3 def fun2():  
    print('fun2 호출!')  
    fun3()  
4 def fun3():  
    print('fun3 호출!')  
1 fun1()
```

fun1 호출!
fun2 호출!
fun3 호출!

또 다른 함수 호출

pass 사용

• pass 사용

- pass를 이용해서 실행문을 생략할 수 있다.

```
def printTodayWeather():  
    pass  
  
def printTomorrowWeather():  
    pass  
  
printTodayWeather()  
printTomorrowWeather()
```

실행문 없음

실행문 없음

• 실습

또 다른 함수 호출
pass 사용

➤ 다음과 같이 구구단 출력 함수가 연쇄적으로 호출되도록 함수를 선언해보자.

```
def guguDan2():
    for i in range(1, 10):
        print('2 * {} = {}'.format(i, 2*i))

    guguDan3()

def guguDan3():
    for i in range(1, 10):
        print('3 * {} = {}'.format(i, 3*i))

    guguDan4()

def guguDan4():
    for i in range(1, 10):
        print('4 * {} = {}'.format(i, 4*i))

    guguDan5()

def guguDan5():
    for i in range(1, 10):
        print('5 * {} = {}'.format(i, 5*i))

guguDan2()
```

2 * 1 = 2	3 * 1 = 3
2 * 2 = 4	3 * 2 = 6
2 * 3 = 6	3 * 3 = 9
2 * 4 = 8	3 * 4 = 12
2 * 5 = 10	3 * 5 = 15
2 * 6 = 12	3 * 6 = 18
2 * 7 = 14	3 * 7 = 21
2 * 8 = 16	3 * 8 = 24
2 * 9 = 18	3 * 9 = 27
	↓
5 * 1 = 5	4 * 1 = 4
5 * 2 = 10	4 * 2 = 8
5 * 3 = 15	4 * 3 = 12
5 * 4 = 20	4 * 4 = 16
5 * 5 = 25	4 * 5 = 20
5 * 6 = 30	4 * 6 = 24
5 * 7 = 35	4 * 7 = 28
5 * 8 = 40	4 * 8 = 32
5 * 9 = 45	4 * 9 = 36