

# 파이썬

## 18강. 중첩함수

# 1. 중첩함수

- 중첩함수는 함수 내부에 또 다른 함수가 내장된 형태를 의미한다. 내부 함수를 포함하는 함수를 외부 함수라고 표현하겠다. 다음은 중첩함수의 형식을 나타내고 있다. 내부 함수도 일반 함수와 동일하게 인 수와 return 명령문으로 갖는다.

```
def 외부함수(인수) : *  
    실행문 *  
    def 내부함수(인수) : *  
        실행문 *  
        return 값 *  
    return 내부함수 *
```

## 2. 일급함수와 함수 클로저

- 파이썬의 중첩함수는 외부함수나 내부함수를 변수에 저장할 수 있는데, 이러한 특성을 갖는 함수를 일 급함수(First class Function)라고 한다.
- 특히 내부함수는 외부함수의 return 명령문을 이용하여 반환하는 형태를 함수 클로저(Function cloure)라고 한다. 함수 클로저는 외부함수가 종료되어도 내부 함수에서 선언된 변수가 메모리에서 소멸되지 않은 상태로 내부 함수를 활용할 수 있다.

## 2. 일급함수와 함수 클로저

chapter05.lecture.step04\_inner\_func.py ↗

Python Console ↗

```
# (1) 일급 함수 ↗
def a(): # outer ↗
    print('a 함수') ↗
    def b(): # inner ↗
        print('b 함수') ↗
    return b ↗
b = a() # 외부 함수 호출 : a 함수 ↗
b() # 내부 함수 호출 : b 함수 ↗

# (2) 함수 클로저 ↗
data = list(range(1, 101)) ↗
def outer_func(data): ↗
    dataSet = data # 값(1~100) 생성 ↗
    # inner ↗
    def tot(): ↗
        tot_val = sum(dataSet) ↗
        return tot_val ↗
    def avg(tot_val): ↗
        avg_val = tot_val / len(dataSet) ↗
        return avg_val ↗
    return tot, avg # inner 반환 ↗

# 외부 함수 호출 : data 생성 ↗
tot, avg = outer_func(data) ↗

# 내부 함수 호출 ↗
tot_val = tot()
print('tot =', tot_val)
avg_val = avg(tot_val)
print('avg =', avg_val) ↗
```

a 함수 ↗  
b 함수 ↗  
tot = 5050 ↗  
avg = 50.5 ↗

## 2. 일급함수와 함수 클로저

- 일급함수와 함수 클로저 예
- # (1) 일급함수
- 함수 a()를 호출하면 내부함수 b()가 반환된다. 이렇게 반환된 함수를 변수 b에 저장한다. 즉 함수를 객체로 만들어서 사용하는 것을 일급함수라고 한다.
- # (2) 함수 클로저
- 외부함수에 의해서 반환된 tot, avg 함수는 외부함수가 종료되어도 객체로 만들어지기 때문에 합계와 평균을 계산하는데 이용할 수 있다. 이러한 함수를 함수 클로저라고 한다.

### 3. 중첩함수 역할

- 중첩함수에서 외부 함수와 내부 함수는 다음과 같은 일정한 역할을 담당한다.
- 외부 함수 : 함수에서 사용할 자료를 만들고, 내부 함수를 포함하는 역할을 한다.
- 내부 함수 : 외부 함수에서 만든 자료를 연산하고 조작하는 역할을 한다.
- 다음은 외부 함수에서 데이터 셋을 만들고, 내부 함수에서 데이터 셋의 분산과 표준편차를 구하는 예문 이다.

### 3. 중첩함수 역할

chapter05.lecture.step04\_inner\_func.py ↵

Python Console ↵

```
from statistics import mean # 평균 ↵
from math import sqrt # 제곱근 ↵

data = [4, 5, 3.5, 2.5, 6.3, 5.5] ↵

# (1) 외부 함수 : 산포도 함수 ↵
def scattering_func(data): # outer
    dataSet = data # data 생성 ↵

    # (2) 내부 함수 : 산술평균 반환 ↵
    def avg_func(): ↵
        avg_val = mean(dataSet)
        return avg_val ↵

    # (3) 내부 함수 : 분산 반환 ↵
    def var_func(avg): ↵
        diff = [ (data - avg) ** 2 for data
in dataSet ] ↵
        print(sum(diff)) # 차의 합 ↵
        var_val = sum(diff) / (len(dataSe↵
t) - 1) ↵
        return var_val ↵
```





### 3. 중첩함수 역할

- 산포도를 구하는 중첩함수 예
- # (1) 외부 함수 : 산포도 함수
- 외부 함수는 6개의 원소를 갖는 data 변수를 인수로 받아서 dataSet 변수에 초기화하여 산포도를 구하기 위한 자료를 생성한다. 또한 산술평균과 분산 그리고 표준편차를 계산하는 3개의 내부 함수를 포함하고, 이들을 return으로 반환하는 역할을 한다.
- # (2) 내부 함수 : 산술평균 반환
- 외부함수에서 만든 dataSet을 이용하여 산술평균을 반환하는 함수이다.
- # (3) 내부 함수 : 분산 반환
- 산술평균을 인수로 받아서 dataSet의 각 변량과 차의 제곱을 계산하여 순서대로 리스트 변수인 diff
- 에 추가한다. 그리고 변량의 길이에서 1을 뺀 값으로 나누어서 표본의 분산을 계산하고 반환한다.

### 3. 중첩함수 역할

- # (4) 내부 함수 : 표준편차 반환
- 분산을 인수로 받아서 양의 제곱근을 적용하여 표준편차를 계산하고 반환한다.
- # (5) 내부 함수 호출
- 반환된 함수 클로저를 이용하여 내부함수를 직접 호출한다. 분산은 산술평균을 실인수로 넘겨줘야하기 때문에 avg()의 반환값을 var() 함수의 실인수로 넣어주고, 표준편차는 분산을 실인수로 넘겨줘야하기 때문에 var(avg()))의 반환값을 std() 함수의 실인수로 넣어준다.

## 4. 획득자, 지정자, nonlocal

- 내부 함수는 중첩함수에서 함수 클로저 역할도 하지만 함수내의 자료를 외부로 획득하거나 자료를 수정하는 역할도 한다. 이러한 내부 함수를 획득자(Getter)와 지정자(Setter)라고 부르며, 역할과 필수 요건은 다음과 같다.
- 획득자 함수 : 함수 내부에서 생성한 자료를 외부로 반환하는 함수로 반드시 return 명령문을 갖는다.
- 지정자 함수 : 함수 내부에서 생성한 자료를 외부에서 수정하는 함수로 반드시 매개변수를 갖는다. 만약 외부 함수에서 생성된 자료를 수정할 경우에는 해당 변수에 nonlocal 명령어를 쓴다.
- 특히 지정자 함수(setter function)에서 nonlocal 명령어는 내부 함수에서 외부 함수의 변수를 사용 할 경우 변수 앞에 붙인다. nonlocal 명령어 형식은 다음과 같다.

## 4. 획득자, 지정자, nonlocal

```
def 외부 함수 () :  
    변수명 = 값  
    def 내부 함수 () :  
        nonlocal 변수명
```

## 4. 획득자, 지정자, nonlocal

chapter05.lecture.step04\_inner\_func.py ↵

Python Console ↵

```
# (1) 중첩함수 정의 ↵
def main_func(num): ↵
    num_val = num # 자료 생성 ↵
    def getter(): # 획득자 함수, return 있음 ↵
        return num_val ↵
    def setter(value): # 지정자 함수 인수 있음 ↵
        nonlocal num_val # nonlocal 명령어 ↵
        num_val = value ↵

    return getter, setter # 함수 클로저 반환 ↵

# (2) 외부 함수 호출 ↵
getter, setter = main_func(100) # num 생성 ↵

# (3) 획득자 호출 ↵
print('num =', getter()) # 획득한 num 확인. ↵
num = 100 ↵

# (4) 지정자 획득 ↵
setter(200) # num 값 수정 ↵
print('num =', getter()) # num 수정 확인 ↵
num = 200 ↵
```

## 4. 획득자, 지정자, nonlocal

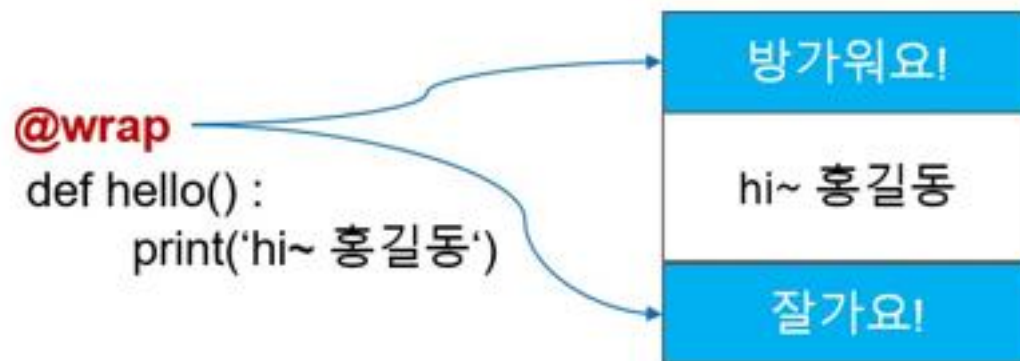
- 획득자와 지정자 예
- # (1) 중첩함수 정의
- 외부 함수에서 num을 인수로 받아서 num\_val 변수에 할당하여 자료를 만든다. 또한 획득자와 지정자 함수를 포함하고, 이들을 return으로 반환하는 역할을 한다.
- # (2) 외부 함수 호출
- 실인수 100을 넘겨서 중첩함수를 호출하면 num\_val 변수에 100이 할당되고, 두 개의 함수 클로저를 반환한다.
- # (3) 획득자 호출
- 함수 클로저 getter를 이용하여 획득자 함수를 호출하면 num\_val의 값이 반환된다.

## 4. 획득자, 지정자, nonlocal

- # (4) 지정자 획득
- 함수 클로저 setter를 이용하여 실인수 200으로 지정자 함수를 호출하면 외부 함수에서 만들어진 num\_val의 값이 200으로 수정된다. 이때 num\_val 변수가 내부 함수의 변수가 아니고 외부 함수에서 만들어진 변수라는 의미로 nonlocal 명령어를 변수 앞부분에 붙인다. 만약 명령어를 생략하면 내부 함수의 변수로 인식되어 num\_val 변수의 값이 수정되지 않는다.

## 5. 함수 장식자

- 함수 장식자(decoration)는 기존 함수의 시작 부분과 종료 부분에 기능을 장식해서 추가해 주는 별도의 함수를 의미한다.
- 즉, 현재 실행되는 함수를 파라미터로 받아서 꾸며줄 내용과 함께 해당 함수를 감싸주는 함수(wrapping function)이다.
- hello() 함수를 @wrap 함수 장식자에 의해서 래핑(wrapping)하는 과정을 나타내고 있다.





## 5. 함수 장식자

- 함수 장식자는 @함수명 형태로 함수 앞에 '@' 기호를 붙인다. 함수 장식자 역시 별도의 함수로 존재하며 적용할 일반함수 앞부분에 작성하면 해당 함수를 인수로 받아서 처리하는 역할을 한다.

```
@함수 장식자↵  
def 함수명 () :↵  
    실행문↵
```

## 5. 함수 장식자

chapter05.lecture.step04\_inner\_func.py ↵

Python Console ↵

```
# (1) 래퍼 함수 ↵
def wrap(func) : ↵
    def decorated() :
        print('방가워요!') # 시작 부분에 삽입
        func() # 인수로 넘어온 함수(hello)
        print('잘가요!') # 종료 부분에 삽입 ↵
    return decorated # 클로저 함수 반환 ↵
```

```
# (2) 함수 장식자 적용 ↵
@wrap ↵
def hello() : ↵
    print('hi ~ ', "홍길동") ↵
```

```
# (3) 함수 호출 ↵
hello() ↵
```

방가워요! ↵  
hi ~ 홍길동  
잘가요! ↵

## 5. 함수 장식자

- 해설 함수 장식자 예
- # (1) 래퍼 함수
- 함수 장식자로 사용할 함수를 선언하고 있다. (func)는 현재 실행되는 함수(hello)를 인수로 받는 매개변수이고, decorated() 내부 함수는 인수로 받는 함수의 시작 부분과 종료 부분에 삽입할 명령문을 작성하고 있다. 인수로 넘어오는 함수는 func() 형태로 호출한다. 끝으로 내부 함수를 반환한다.
- # (2) 함수 장식자 적용
- hello()함수를 대상으로 래퍼 함수를 적용하기 위해서 함수 장식자 (@wrap)를 함수 앞부분에 붙인다.
- # (3) 함수 호출
- 함수 장식자가 적용된 hello() 함수를 호출하면 함수 장식자가 호출되면서 hello() 함수 실행문('hi ~ 홍길동')을 중심으로 시작 부분에 '방가워요!', 종료 부분에 '잘가요!' 가 각각 장식으로 추가된다.

**THANK YOU**