

# chapter 10. 함수 만들기

## 1) 함수 만들기 - input

① 함수 hello\_world를 만들어 봅시다.

- 이름 값을 받는 입력 매개변수 name(문자)
- 'Hello '와 ' , welcome to Python World!' 사이에 name을 넣어 출력
- (Hello name, welcome to Python World! 형태)
- name에 값을 지정하여 함수의 결과를 확인하시오.

```
hello_world('Ann')
-----
Hello Ann, welcome to Python World!
```

In [ ]:

```
def hello_world(name) :
    print('Hello ' + name + ', welcome to Python World!')

hello_world('Ann')
```

Hello Ann, welcome to Python World!

② 함수 say\_hey를 만들어 봅시다.

- 횟수를 받는 입력 매개변수 count(숫자)
- 입력받은 count만큼 'Hey!'를 출력
- count에 값을 지정하여 함수의 결과를 확인하시오.

```
say_hey(5)
-----
Hey! Hey! Hey! Hey! Hey!
```

In [ ]:

```
def say_hey(count) :
    print('Hey! ' * count)

say_hey(5)
```

Hey! Hey! Hey! Hey! Hey!

③ 함수 hello를 만들어 봅시다.

- 이름 값을 받는 입력 매개변수 name(문자), 숫자를 받는 loud(숫자) (이때, loud = 1로 선언)
- loud = 1일 때, 'HELLO '와 '!!!' 사이에 name(대문자)을 넣어 출력
- loud = 0이면, 'Hello '와 '.' 사이에 name(소문자)을 넣어 출력
- name, loud에 값을 지정하여 함수의 결과를 확인하시오.
- name에만 값을 지정하여 함수의 결과를 확인하시오.

```
hello(name = 'Kim', loud = 0)
hello(name = 'Kim')
-----
Hello kim.
HELLO KIM!!
```

In [ ]:

```
def hello(name, loud = 1) :
```

```

if loud == 1 :
    print('HELLO ' + name.upper() + '!!')
else :
    print('Hello ' + name.lower() + '!.')

hello(name = 'Kim', loud = 0)
hello(name = 'Kim')

```

Hello kim.  
HELLO KIM!!

④ 함수 add를 만들어 봅시다.

- 계산할 숫자들을 받는 입력 매개변수 a(숫자), b(숫자)
- a와 b의 합을 변수 sum에 저장
- '덧셈 결과는' 뒤에 sum의 값을 넣어서 출력 (덧셈 결과는 sum 형태)
- a, b에 값을 지정하여 함수의 결과를 확인하시오.

```

add(7, 2)
-----
덧셈 결과는 9

```

```

In [ ]: def add(a, b) :
        sum = a + b
        print('덧셈 결과는 ', sum)

add(7, 2)

```

덧셈 결과는 9

⑤ 함수 numbers를 만들어 봅시다.

- 출력할 숫자들을 받는 입력 매개변수 nums(숫자)는 입력 값의 수를 정하지 않고 들어오는 대로 받음
- 입력받은 숫자들을 출력
- nums에 값을 지정하여 함수의 결과를 확인하시오.
- 힌트 : 입력받을 매개변수의 수를 지정하지 않고 사용하려면 \*nums

```

numbers(2, 4, 6, 8, 10)
-----
(2, 4, 6, 8, 10)

```

```

In [ ]: def numbers(*nums) :
        print(nums)

numbers(2, 4, 6, 8, 10)

```

(2, 4, 6, 8, 10)

⑥ 함수 average를 만들어 봅시다.

- 평균을 구할 수들을 받는 입력 매개변수 nums(숫자)는 입력 값의 수를 정하지 않고 들어오는 대로 받음
- 입력받은 숫자들의 평균을 변수 avg에 저장하여 출력
- nums에 값을 지정하여 함수의 결과를 확인하시오.

```

average(2, 4, 6, 8, 10)
-----

```

6.0

```
In [ ]: def average(*nums) :
        avg = sum(nums) / len(nums)
        print(avg)

        average(2, 4, 6, 8, 10)
```

6.0

## 2) 함수 만들기 - output

① 함수 calculate1을 만들어 봅시다.

- 입력 매개변수 : 숫자1, 숫자2, 사칙연산 부호(+,-,\*,/)
- 숫자1과 숫자2 사이에 사칙연산자 부호에 해당하는 연산을 수행후
- 결과를 print로 출력합니다.

```
calculate1(11, 2, '*')
-----
22
```

```
In [ ]: def calculate1(n1, n2, operator) :
        if operator == '+' :
            result = n1 + n2
        elif operator == '-' :
            result = n1 - n2
        elif operator == '*' :
            result = n1 * n2
        else :
            result = n1 / n2

        print(result)

        calculate1(11, 2, '*')
```

22

② 위에서 만든 calculate1의 결과를 변수 c로 저장해 봅시다.

- 그런 다음 c를 출력해 봅시다.
- 결과가 담기나요?

```
In [ ]: c = calculate1(11, 2, '*')
        print(c)
```

22

None

③ 위에서 만든 calculate1의 결과를 변수 c로 저장하기 위해서

- 결과를 반환하도록 코드를 수정하세요.
- 힌트 : return
- 이제 다시 결과를 c로 저장하고 출력합니다.
- 결과가 담기나요?

```
In [ ]: def calculate1(n1, n2, operator) :
        if operator == '+' :
```

```

        result = n1 + n2
    elif operator == '-' :
        result = n1 - n2
    elif operator == '*' :
        result = n1 * n2
    else :
        result = n1 / n2

    return result

c = calculate1(11, 2, '*')
print(c)

```

22

④ 함수 calculate2 를 생성합니다.

- 숫자 두개를 받아서
- 몫과 나머지를 계산하고
- 두 결과를 반환해야 합니다.
- 함수 실행 후 결과를 r1, r2에 저장한 후 출력합니다.

```

r1, r2 = calculate2(11, 2)
print(r1, r2)
-----
5 1

```

In [ ]:

```

def calculate2(n1, n2) :
    return n1//n2, n1% n2

r1, r2 = calculate2(11, 2)
print(r1, r2)

```

5 1

⑤ [심화] 함수 even을 만들어 봅시다.

- 입력 매개변수로 리스트를 입력받아.
- 입력받은 리스트의 값들 중 짝수로만 구성된 새로운 리스트 출력
- 힌트: 함수 처리 코드 내 짝수를 저장할 빈 리스트 result 생성

```

even([2, 3, 6, 8, 11])
-----
[2, 6, 8]

```

In [ ]:

```

def even(num_list) :
    result =[]
    for i in num_list :
        if i % 2 == 0 :
            result.append(i)
    return result

even([2, 3, 6, 8, 11])

```

Out[ ]: [2, 6, 8]

⑥ [심화] 복리 계산기 함수 interest를 만들어 봅시다.

- 입력변수 : 예금액, 이자, 연수

- 결과 : 복리로 불어난 예금액

```
result = interest(120000, 0.05, 5)
print(result)
-----
145860.75
```

In [ ]:

```
def interest(deposit, rate, years) :
    balance = deposit
    for i in range(1, years) :
        balance *= (1 + rate)
    return balance

result = interest(120000, 0.05, 5)
print(result)
```

145860.75

### 3) 함수 만들기 - 예외처리

- ① 함수 divide를 만들어 봅시다.

- 입력변수 : num1, num2
- 처리 : num1 / num2
- 결과 : 나눈 결과를 반환
- 함수를 만들고 숫자 두개를 넣고 결과를 확인해 봅시다.

In [ ]:

```
def divide(num1, num2) :
    result = num1 / num2
    return result

divide(10, 2)
```

Out[ ]: 5.0

- ② 위 문제에서 만약 num2 에 0을 넣으면 오류가 발생합니다.

- ZeroDivisionError: division by zero
- 이를 해결하기 위해 오류 처리 코드를 넣어 봅시다.
- num2에 0이 들어오면, "0으로 나눌 수 없습니다." 라고 출력하고,
- num2에 다른 숫자가 들어오면, 나눈 결과를 반환하시오.

```
divide(10, 0)
-----
0으로 나눌 수 없습니다.
```

In [ ]:

```
def divide(num1, num2) :
    if num2 == 0 :
        print('0으로 나눌 수 없습니다.')
    else:
        result = num1 / num2
        return result

divide(10, 0)
```

0으로 나눌 수 없습니다.

③ 위 결과에 이어서, num1, num2 모두 int 인지 확인하는 코드도 작성합니다.

- 힌트 : 정수인지 확인 num1 is int

```
divide(10, 'a')
```

-----

입력값이 모두 숫자여야 합니다.

In [ ]:

```
def divide(num1, num2) :  
    if num2 == 0 :  
        print('0으로 나눌 수 없습니다.')  
    elif (num1 is not int) or (num2 is not int) :  
        print('입력값이 모두 숫자여야 합니다.')  
    else:  
        result = num1 / num2  
        return result
```

```
divide(10, 'a')
```

입력값이 모두 숫자여야 합니다.