

05 함수

참고자료: <https://wikidocs.net/24> (<https://wikidocs.net/24>)

1. 파이썬 함수 구조

- 함수 만들기

```
def 함수명(매개변수):  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
    return 리턴값
```

- 함수 사용:

```
print(add(3, 4))
```

- 매개변수:** 함수에 전달된 값을 저장하는 변수
- 인수:** 함수에 전달하는 값

In [1]:

```
def add(a, b):  
    return a + b
```

In [2]:

```
a, b = 3, 4  
c = add(a, b)  
print(c)
```

7

In [4]:

```
print(add(a, b))
```

7

1-1. 입력값이 없는 함수

In [7]:

```
def say():  
    return 'Hi'
```

In [8]:

```
say()
```

Out[8]:

'Hi'

- ★ 주피터 노트북이라서 리턴값이 보임. vscode에서는 리턴값이 보이지 않음
- ▶ 변수에 담아 프린터를 하거나 프린트를 해야함

In [10]:

```
s = say()  
print(s)
```

Hi

In [11]:

```
print(say())
```

Hi

1-2. 리턴이 없는 경우

In [5]:

```
def say(a):  
    print(a, '를 프린트 합니다.')
```

In [6]:

```
say(2)
```

2 를 프린트 합니다.

1-3. 입력값도 리턴값도 없는 경우

In [12]:

```
def say():  
    print('Hi python')
```

In [13]:

```
say()
```

Hi python

1-4. 리턴 값은 하나!

리턴을 두개 할 경우 위에게 위주로 나타난다.

리턴 값이 두개 필요할 때는 return을 두 번 하는 것이 아닌 return a, b 을 사용한다. 이때 튜플 형태로 반환된다.

In [32]:

```
def cal(num1, num2):  
    return num1 + num2  
    return num2 - num2
```

In [33]:

```
cal(3, 4)
```

Out[33]:

7

In [34]:

```
def cal(num1, num2):  
    return num1 + num2, num2 - num2
```

In [35]:

```
cal(3, 4)
```

Out[35]:

(7, 0)

2. 매개변수에 기본값 지정하기

2-1. 매개변수에 기본값 지정하는 경우

In [16]:

```
def average1(num1, num2):  
    result = (num1 + num2)/2  
    return result
```

In [22]:

```
average1(0,10)
```

Out[22]:

5.0

In [18]:

```
average1()
```

```
-----  
-----  
TypeError                                Traceback (most recent call last)  
~WAppDataWLocalWTempWipykernel_8896W166441766.py in <module>  
----> 1 average1()
```

TypeError: average1() missing 2 required positional arguments: 'num1' and 'num2'

2-2. 기본값을 지정한 매개변수 경우

In [14]:

```
def average2(num1=5, num2=7):  
    result = (num1 + num2)/2  
    return result
```

In [19]:

```
average2() # num1 = 5, num2 = 7
```

Out[19]:

6.0

In [20]:

```
average2(7) # num1 = 7, num2 = 7
```

Out[20]:

7.0

In [21]:

```
average2(0,10) # num1 = 0, num2 = 10
```

Out[21]:

5.0

2-3. 기본값을 지정한 매개변수와 지정하지 않는 매개변수를 혼합하는 경우

In [23]:

```
def average3(num1, num2=7):  
    result = (num1 + num2)/2  
    return result
```

In [24]:

```
average3() # num1 = 5, num2 = 7
```

```
-----  
-----  
TypeError                                 Traceback (most recent call last)  
~WAppDataWLocalWTempWipykernel_8896W3692937726.py in <module>  
----> 1 average2() # num1 = 5, num2 = 7  
  
TypeError: average2() missing 1 required positional argument: 'num1'
```

In [27]:

```
average3(6) # num1 = 6, num2 = 7
```

Out[27]:

6.5

In [28]:

```
average3(0,10) # num1 = 0, num2 = 10
```

Out[28]:

5.0

※ 주의할 점!!

기본값을 넣지 않은 매개변수는 기본값을 지정한 매개변수 앞에 있어야한다.

```
def func(num, num2 = a)
```

In [29]:

```
def average4(num1 = 7, num2):  
    result = (num1 + num2)/2  
    return result
```

File "C:\Users\User\AppData\Local\Temp\ipykernel_8896\1455197802.py", line 1

```
def average4(num1 = 7, num2):  
    ^
```

SyntaxError: non-default argument follows default argument

3. 가변인자

In [36]:

```
def add_many(*args):  
    result = 0  
    for i in args:  
        result = result + i  
    return result
```

In [66]:

```
add_many()
```

Out[66]:

0

In [37]:

```
add_many(1, 2, 3)
```

Out[37]:

6

In [38]:

```
add_many(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

Out[38]:

55

3-1. 가변인자 위치

```
def 함수이름(매개변수, 가변인수, 매개변수 = 기본값):  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
    return
```

In [80]:

```
def add_many2(num1, num2 = 5, *args):  
    print('num1', num1)  
    print('num2', num2)  
    print('args', args)
```

In [81]:

```
add_many2(1)
```

```
num1 1  
num2 5  
args ()
```

In [82]:

```
add_many2(1,2,3,4,5,6,7,8,9,10)
```

```
num1 1  
num2 2  
args (3, 4, 5, 6, 7, 8, 9, 10)
```

In [83]:

```
def add_many2(num1, *args, num2 = 5, ):  
    print('num1', num1)  
    print('num2', num2)  
    print('args', args)
```

In [84]:

```
add_many2(1,2,3,4,5,6,7,8,9,10)
```

```
num1 1  
num2 5  
args (2, 3, 4, 5, 6, 7, 8, 9, 10)
```

4. 키워드 파라미터

In [85]:

```
def func_param_with_kwargs(**kwargs):  
    print("kwargs=",end=""), print(kwargs)
```

In [86]:

```
func_param_with_kwargs(mobile="01012341234", address="seoul")
```

```
kwargs={'mobile': '01012341234', 'address': 'seoul'}
```

4-1. 키워드 파라미터 위치

In [93]:

```
def func_param_with_kwargs(name, age, address=0, **kwargs):  
    print("name=",end=""), print(name)  
    print("age=",end=""), print(age)  
    print("address=",end=""), print(address)  
    print("kwargs=",end=""), print(kwargs)  
  
func_param_with_kwargs("정우성", "20", mobile="01012341234", address="seoul")
```

```
name=정우성  
age=20  
address=seoul  
kwargs={'mobile': '01012341234'}
```

In [96]:

```
def func_param_with_kwargs(name, age, **kwargs, address=0):  
    print("name=",end=""), print(name)  
    print("age=",end=""), print(age)  
    print("address=",end=""), print(address)  
    print("kwargs=",end=""), print(kwargs)  
  
func_param_with_kwargs("정우성", "20", mobile="01012341234", address="seoul")
```

File "C:\Users\User\AppData\Local\Temp\ipykernel_8896\2133418597.py", line 1

```
def func_param_with_kwargs(name, age, **kwargs, address=0):  
    ^
```

SyntaxError: invalid syntax

In [94]:

```
def mixed_params(age, name="아이유", *args, address, **kwargs):  
    print("name=",end=""), print(name)  
    print("args=",end=""), print(args)  
    print("age=",end=""), print(age)  
    print("kwargs=",end=""), print(kwargs)  
    print("address=",end=""), print(address)  
  
mixed_params(20, "정우성", "01012341234", "male", mobile="01012341234",  
address="seoul")
```

```
name=정우성  
args=('01012341234', 'male')  
age=20  
kwargs={'mobile': '01012341234'}  
address=seoul
```

In []: