함수

Index

함수?

- 반복적으로 사용되는 가치 있는 부분
- 흐름 파악

파이썬 함수의 구조

```
def 함수_이름(매개변수):    # def : 함수를 만들 때 사용하는 예약어
  수행할_문장1
  수행할_문장2
  ...
```

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

```
a = 3
b = 4
c = add(a, b)   # add(3, 4)의 리턴값을 c에 대입
print(c)
>>> 7
```

매개변수와 인수

- 매개변수 : 함수에 입력으로 전달된 값을 받는 변수
- 인수 : 함수를 호출할 때 전달하는 입력값

```
def add(a, b):   # a, b는 매개변수
  return a+b

print(add(3, 4))   # 3, 4는 인수
```

입력값과 리턴값에 따른 함수의 형태

일반적인 함수

```
def 함수_이름(매개변수):
  수행할_문장
  ...
  return 리턴값
```

```
리턴값을_받을_변수 = 함수_이름(입력_인수1, 입력_인수2, ...)
```

입력값이 없는 함수

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

```
a = say()
print(a)
>>> Hi
```

```
리턴값을_받을_변수 = 함수_이름()
```

리턴값이 없는 함수

```
def add(a, b):
  print("%d, %d의 합은 %d입니다." % (a, b, a+b))
```

```
add(3, 4)
>>> 3, 4의 합은 7입니다.
```

```
함수_이름(입력_인수1, 입력_인수2, ...)
```

입력값도, 리턴값도 없는 함수

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

```
say()
>>> Hi
```

```
함수_이름()
```

매개변수를 지정하여 호출하기

```
def sub(a, b):
  return a - b
```

```
result = sub(a=7, b=3)   # a에 7, b에 3을 전달
print(result)
>>> 4
```

입력값이 몇 개가 될지 모를 때는 어떻게 해야 할까?

```
def 함수_이름(*매개변수):
  수행할_문장
  ...
```

여러 개의 입력값을 받는 함수 만들기

```
def add_many(*args):
  result = 0
  for i in args:
    result = result + i   # *args에 입력받는 모든 값을 더한다.
  return result
```

```
result = add_many(1, 2, 3)
print(result)
>>> 6
result = add_many(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(result)
>>> 55
```

키워드 매개변수, kwargs

```
def print_kwargs(**kwargs):
  print(kwargs)
```

```
print_kwargs(a=1)
>>> {'a': 1}
print_kwargs(name='foo', age=3)
>>> {'age': 3, 'name': 'foo'}
```

함수의 리턴값은 언제나 하나이다

```
def add_and_mul(a,b):
  return a+b, a*b
```

```
result = add_and_mul(3,4)
>>> result = (7,12)
```

```
result1, result2 = add_and_mul(3, 4)
>>> result1 = 7, result2 = 12
```

```
def add_and_mul(a,b):
  return a+b
  return a*b
```

```
result = add_and_mul(2, 3)
print(result)
>>> 5
```

```
def say_nick(nick):
  if nick == "바보":
    return
  print("나의 별명은 %s 입니다." % nick)
```

매개변수에 초깃값 미리 설정하기

```
def say_myself(name, age, man=True):
    print("나의 이름은 %s 입니다." % name)
    print("나이는 %d살입니다." % age)
    if man:
        print("남자입니다.")
    else:
        print("여자입니다.")
```

```
say_myself("박응선", 27, False) >>> ... 여자입니다.
```

> 초깃값이 있는 매개변수를 초깃값이 없는 매개변수 앞에 사용하면 안 된다.

함수 안에서 선언한 변수의 효력 범위

```
a = 1
def vartest(a):
  a = a + 1

vertest(a)
print(a)

>>> 1
```

```
def vartest(a):
    a = a + 1

vartest(3)
print(a)

>>> Error   # main에 a가 없어서
```

함수 안에서 함수 밖의 변수를 변경하는 방법

return 사용하기

```
a = 1
def vartest(a):
  a = a + 1
  return a

a = vartest(a)
print(a)
>>> 2
```

global 명령어 사용하기

```
a = 1
def vartest():
  global a
  a = a+1

vartest()
print(a)
>>> 2
```

- 그다지 좋은 방법은 아님

lambda 예약어

- 함수를 생성할 때 사용하는 예약어
- 간결함, 복잡하지 않거나 def를 사용할 수 없을 때 사용

```
함수_이름 = lamba 매개변수1, 매개변수2, ... : 매개변수를_이용한_표현식
```

```
add = lamba a, b: a+b
result = add(3, 4)
print(result)
>>> 7
```