Ch9. Function

Function

```
a = 5 + 3
print('a=', a)

....

b = 6+4
print('b=', b)
```

Repeat same processes!

f(x, y) = x + y

Function

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Defining a function

```
def function_name (arg1, arg2) :
    statement1
    statement2
    return
```

Defining a function

```
a = 5 + 3
print('a=', a)

....

b = 6+4
print('b=', b)
```

f(x, y) = x + y

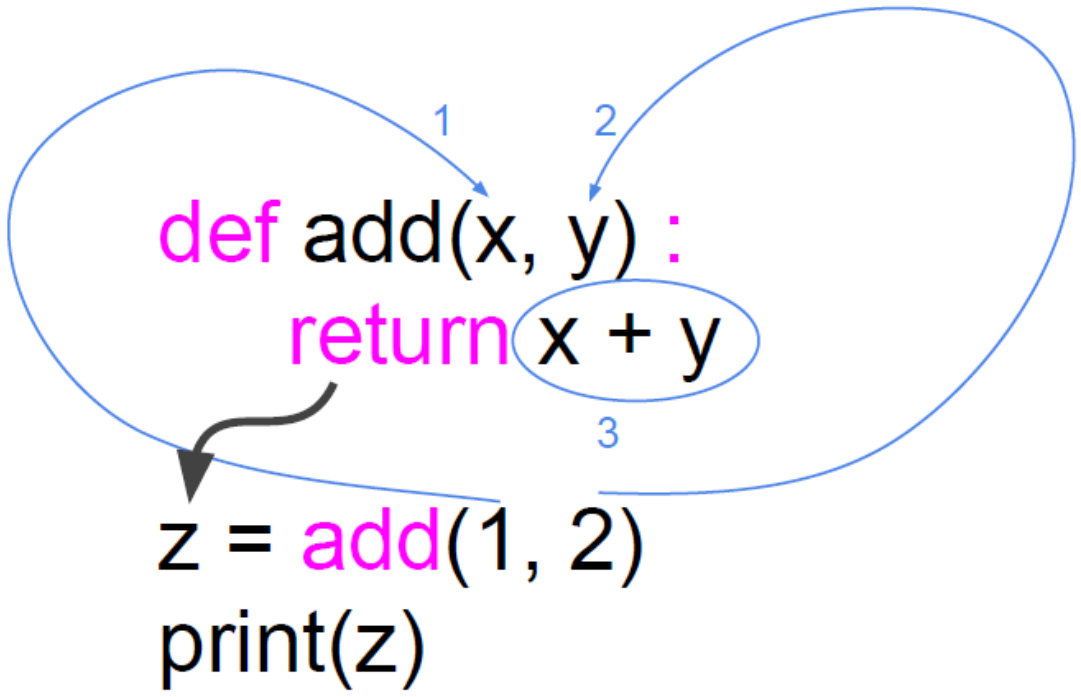
```
def add(x, y) :
    return x + y
```

Defining a function

```
def add(x, y) :
    return x + y
```

Define function **add** to gets two **arguments** x, y.
Function **add returns** the result of x+y

Calling a function

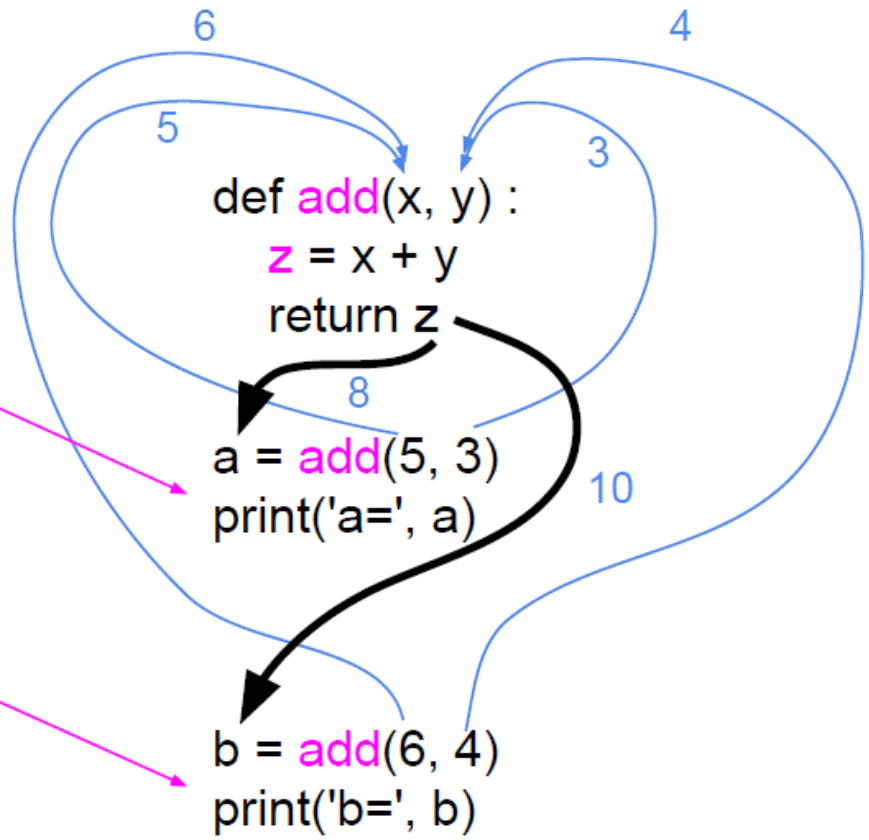


Function

```
a = 5 + 3
print('a=', a)

....

b = 6+4
print('b=', b)
```



Function without arguments

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

a = say()
print(a)
```

Function without return value

```
def add(x, y) :
    print('value=', x+y)
    return

add(x, y)
```

Function without arguments/return value

```
def say() :
    print('hi')
    return

say()
```

Function with multiple return values

```
def sum_and_mul(x, y) :
    return x+y, x*y

z = sum_and_mul(3, 4)
print(z)
```

7 12

(7, 12)

Tuple!

Function with multiple return values

```
def sum_and_mul(x, y) :
    return x+y, x*y

sum, mul = sum_and_mul(3, 4)
print(sum)
print(mul)
```

7 12

(7, 12)

Function arguments

Function arguments

Required arguments

Required arguments are the arguments passed to a function in correct positional order. Here, the number of arguments in the function call should match exactly with the function definition.

Function arguments

```
def printme( str ):
    print(str)
    return

printme()
```

```
Traceback (most recent call last):
  File "/Users/woojin/PycharmProjects/hy-cce/week2/psk.py", line 7, in <module>
    printme()
TypeError: printme() missing 1 required positional argument: 'str'

Process finished with exit code 1
```

Function arguments

```
def printme( str ):
    print(str)
    return

printme('my', 'text')
```

```
Traceback (most recent call last):
  File "/Users/woojin/PycharmProjects/hy-cce/week2/psk.py", line 7, in <module>
    printme('my', 'text')
TypeError: printme() takes 1 positional argument but 2 were given

Process finished with exit code 1
```

Function arguments

Keyword arguments

When you use keyword arguments in a function call, the caller identifies the arguments by the parameter name. Python interpreter is able to use the keywords provided to match the values with parameters.

Function arguments

```
def printinfo( name, age ):
    print('Name : ', name)
    print('Age : ', age)
    return

printinfo('Woojin', 3018)
printinfo(age=3018, name='Woojin')
```

Function arguments

Default arguments

A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.

Function arguments

```
def printinfo( name, age=48 ) :
    print('Name : ', name)
    print('Age : ', age)
    return

printinfo('Woojin')
printinfo(age=3018, name='Woojin')
```

전역변수와 지역변수 (Global variable & local variable)

- 전역변수 : 함수 밖에서 선언되며, 프로그램 어디에서든 사용 가능한 변수
- 지역변수 : 함수 안에서 선언되며, 해당 함수 안에서만 사용이 가능한 변수.
  (함수 호출 시 생성되었다가, 함수가 종료되면 변수도 사라진다!)
- 전역변수와 지역변수는 이름이 같더라도 별개의 변수!
- 두 개 이상의 지역변수가 이름이 같더라도, 서로 다른 함수 안에서 선언되었다면 별개의 변수!

전역변수와 지역변수

함수 안

Local variable

Global variable

함수 밖

```
def localVariable():
    var = 'Local'
    print(var)

var = 'Global'
print(var)
localVariable()
print(var)
```

```
Global
Local
Global
```

Assignment6

- Deadline : November 18th
- Upload to portal – Assignment – "Assignment6"
- Upload File Name : assignment6_student ID_name.py (python file) & **Capture result photos**
  - ex : assignment6_2017200966_조수필.py
- If you complete the assignment in class, ask the assistant for confirmation.

Ch9. Function 참고자료

전달인자, 매개변수

① 전달인자 'Meo'를 넣은 함수 Hello 를 실행하면,

② 매개변수 name 에 'Meo'를 대입하여 함수 Hello 를 연산!

```
def Hello(name):
    print('Hello,', name)

Hello('Meo')
Hello('Teo')
Hello('Seo')
Hello('Zeo')
```

```
Hello, Meo
Hello, Teo
Hello, Seo
Hello, Zeo
```

return

```python
def plus(firstnum, secondnum):
    result = firstnum+secondnum
    return result1

firstnum = int(input('첫번째 숫자: '))
secondnum = int(input('두번째 숫자: '))
result2 = plus(firstnum, secondnum)
print('두 수의 합은', result2, '입니다.')
```

```
첫번째 숫자: 3
두번째 숫자: 5
두 수의 합은 8 입니다.
```

return

① plus 함수 안에서 구한 값 result1를 함수 밖에서 사용하고 싶다면, return 을 사용!

② plus 함수의 결과값으로, return1이 반환됨!

③ 이를 result2 라는 전역변수에 저장하여 활용

```
def plus(firstnum, secondnum):
    result = firstnum+secondnum
    return result1

firstnum = int(input('첫번째 숫자: '))
secondnum = int(input('두번째 숫자: '))
result2 = plus(firstnum, secondnum)
print('두 수의 합은', result2, '입니다.')
```

```
첫번째 숫자: 3
두번째 숫자: 5
두 수의 합은 8 입니다.
```