2018 Fall Python Study

5. Function

Function

함수

$$y = x + 1$$

Input

Output

```python
def say_hello():
   # block belonging to the function
   print('hello world')
   # End of function

say_hello() # call the function
say_hello() # call the function again
```

hello world

hello world

```python
def say_hello():
   # block belonging to the function
   print('hello world')
   # End of function

say_hello() # call the function
say_hello() # call the function again
```

같은 작업을 여러 번 진행 할 때

Parameter

매개 변수

```python
def print_max(a, b):
    if a > b:
        print(a, 'is maximum')
    elif a == b:
        print(a, 'is equal to', b)
    else:
        print(b, 'is maximum')
```

```
print_max(5, 10)
```

10 is maximum

```python
def print_max(a, b):
    if a > b:
        print(a, 'is maximum')
    elif a == b:
        print(a, 'is equal to', b)
    else:
        print(b, 'is maximum')
```

```
x = 5
y = 7

print_max(x, y)
```

7 is maximum

```python
def print_max(a, b):
    if a > b:
        print(a, 'is maximum')
    elif a == b:
        print(a, 'is equal to', b)
    else:
        print(b, 'is maximum')
```

Local Variables

```python
x = 50


def func(x):
    print('x is', x)
    x = 2
    print('Changed local x to', x)


func(x)
print('x is still', x)
```

```
x is 50
Changed local x to 2
x is still 50
```

global

```python
x = 50


def func():
    global x

    print('x is', x)
    x = 2
    print('Changed global x to', x)


func()
print('Value of x is', x)
```

```
x is 50
Changed global x to 2
x is still 2
```

```python
x = 50


def func():
    global x

    print('x is', x)
    x = 2
    print('Changed global x to', x)


func()
print('Value of x is', x)
```

Default

```python
def say(message, times=1):
    print(message * times)

say('Hello')
say('World', 5)
```

Hello

WorldWorldWorldWorldWorld

```python
def say(message, times=1):
    print(message * times)

say('Hello')
say('World', 5)
```

Keyword Arguments

```python
def func(a, b=5, c=10):
    print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)
func(25, c=24)
func(c=50, a=100)
```

a is 3 and b is 7 and c is 10

a is 25 and b is 5 and c is 24

a is 100 and b is 5 and c is 50

```python
def func(a, b=5, c=10):
    print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)
func(25, c=24)
func(c=50, a=100)
```

VarArgs parameters

```
def total(a=5, *numbers, **phonebook):
   print('a', a)

   #iterate through all the items in tuple
   for single_item in numbers:
      print('single_item', single_item)

   #iterate through all the items in dictionary
   for first_part, second_part in phonebook.items():
      print(first_part,second_part)

total(10,1,2,3,Jack=1123,John=2231,Inge=1560)
```

```
a 10
single_item 1
single_item 2
single_item 3
Jack 1123
John 2231
Inge 1560
```

return

```python
def maximum(x, y):
    if x > y:
        return x
    elif x == y:
        return 'The numbers are equal'
    else:
        return y

print(maximum(2, 3))
```

3

DocStrings

```python
def print_max(x, y):
    '''Prints the maximum of two numbers.

    The two values must be integers.'''
    # convert to integers, if possible
    x = int(x)
    y = int(y)

    if x > y:
        print(x, 'is maximum')
    else:
        print(y, 'is maximum')

print_max(3, 5)
print(print_max.__doc__)
```

```
5 is maximum
Prints the maximum of two numbers.

The two values must be integers.
```