

Python Programming

함수



1. 함수란

1.1 함수란

관련된 작업을 처리하는데 필요한 문장들을 묶어놓은 것. 필요할 때 마다 함수를 호출해서 사용

1.2 함수 정의

```
def 함수명(파라미터1, 파라미터2, ...):  
    내용  
    내용  
    return 값
```

1.3 함수 호출

```
함수명(파라미터1, 파라미터2, ...)
```



2. 함수

2.1 기본

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
>>>
===== RESTART: D:/workspacePython/function.py =====
오늘은 11월 16일
추워
오늘은 11월 16일
추워
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
# 함수 정의
def printInfo():
    print("오늘은 11월 16일")
    print("추워")

#필요할때마다 호출해서 사용
printInfo()
printInfo()
```

2.2 파라미터(인자) 사용

2.2.1 기본사항

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
>>>
===== RESTART: D:/workspacePython/function.py =====
10 30
x + y = 40
x - y = -20
x * y = 300
x / y = 0.333333
-----
300 10
x + y = 310
x - y = 290
x * y = 3000
x / y = 30.0
-----
55 12
x + y = 67
x - y = 43
x * y = 660
x / y = 4.583333
-----
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
# 함수를 수행하는데 필요한 재료들
def calculate(x, y): # x, y가 파라미터
    print(x, y)
    print("x + y = ", x + y)
    print("x - y = ", x - y)
    print("x * y = ", x * y)
    print("x / y = ", x / y)
    print("-----")

calculate(10, 30) # 함수를 호출할 때 넣어준 값들이 차례대로 파라미터에 들어감
calculate(300, 10)
calculate(55, 12)
```



2.2.2 파라미터 순서 조절

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
30 10
x + y = 40
x - y = 20
x * y = 300
x / y = 3.0
-----
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
def calculate(x, y):
    print(x, y)
    print("x + y = ", x + y)
    print("x - y = ", x - y)
    print("x * y = ", x * y)
    print("x / y = ", x / y)
    print("-----")

calculate(y = 10, x = 30) # 함수를 호출할 때 변수명을 써줘서 순서를 조절 가능
```

2.2.3 기본값 주기

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
100 10
x + y = 110
x - y = 90
x * y = 1000
x / y = 10.0
-----
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
def calculate(x = 300, y = 10): # 파라미터를 선언할 때 값을 넣어주면 기본값이 됨
    print(x, y)
    print("x + y = ", x + y)
    print("x - y = ", x - y)
    print("x * y = ", x * y)
    print("x / y = ", x / y)
    print("-----")

calculate(x = 100) # y값을 넣어주지 않으면 기본값이 들어감
```

2.2.4 가변 파라미터

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
1
<class 'tuple'> (2, 3, 4, 5)
a
<class 'tuple'> ('b', 'c')
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
def printAll(a, *x): # *를 붙여서 파라미터 갯수를 무한으로 받는데 가능, 맨 마지막에 와야함
    print(a)
    print(type(x), x)

printAll(1, 2, 3, 4, 5)
printAll("a", "b", "c")
```

2.2.5 미정의된 파라미터

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
1
<class 'dict'> {'name': '볼펜', 'price': 100}
2
<class 'dict'> {'height': 180, 'weight': 80}
>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
def printAll(a, **x): # **를 붙여서 미정의된 파라미터를 dict형으로 받는데 가능, 맨 마지막에 와야함
    print(a)
    print(type(x), x)

printAll(1, name="볼펜", price=100)
printAll(2, height=180, weight=80)
```



2.3 return

2.3.1 기본 사항

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
30
50
>>> def calculate(x, y):
    hab = x + y
    return hab # 함수 종료, 함수를 수행한 결과값을 함수 호출한 쪽으로 전달

a = calculate(10, 20)
print(a)
b = calculate(30, 20)
print(b)
```

2.3.2 tuple 활용

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
30
50
>>> def calculate(x, y):
    hab = x + y
    cha = x - y
    return hab, cha # tuple형태로 여러 값 리턴 가능

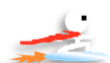
a, b = calculate(10, 20)
print(a)
print(b)
```

2.4 mutable, immutable

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py =====
123
[10, 456, 789]
>>>
def mutableAndImmutable(x, y):
    x = 10
    y[0] = 10

a = 123 # int, float, complex, str, bool은 immutable
b = [123, 456, 789] # 컬렉션 속의 값들은 mutable
mutableAndImmutable(a,b)
print(a) # 값이 바뀌지 않음
print(b) # 값이 바뀜
```





2.6 pass

```
def passTest():  
    # 아무것도 안적어 놓으면 에러  
  
def passTest2():  
    pass # 아무것도 하지 않을때 pass를 사용  
  
print("안녕")
```

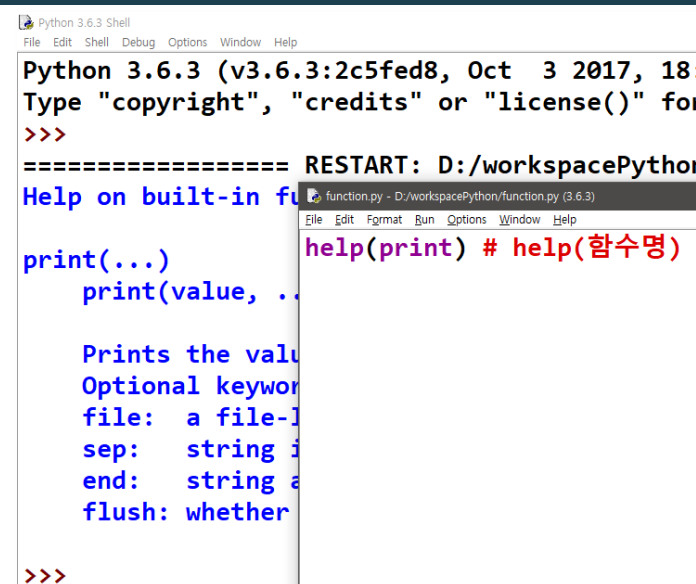
2.7 lambda 함수

```
Python 3.6.3 Shell  
File Edit Shell Debug Options Window Help  
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:/workspacePython/function.py =====  
10  
7  
>>>  
# 이름이 없는 한줄짜리 함수  
# lambda 파라미터1, 파라미터2, ... : 구문(파라미터1, 파라미터2, ...)  
  
print((lambda x, y : x + y)(3, 7))  
  
cha = (lambda x, y : x - y)(10,3)  
print(cha)
```



2.8 문서

2.8.1 도움말 문서보기

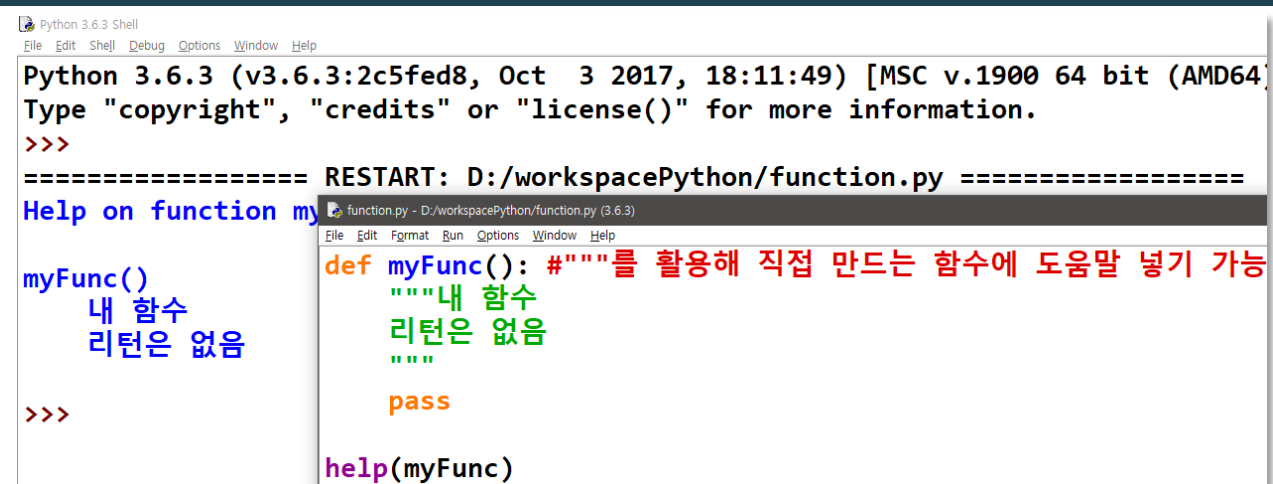


```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more
>>>
===== RESTART: D:/workspacePython/function.py (3.6.3) =====
Help on built-in function print in module builtins:
print(...)
    print(value, ..., sep=' ', end='\\n', file=sys.stdout, flush=False)

    Prints the values to a file-like object.
    Optional keyword arguments:
    file: a file-like object (stream or StreamWriter);
    sep: string to be placed between objects;
    end: string to be placed at the end of the output;
    flush: whether to flush the output buffer.

>>>
```

2.8.2 도움말 만들기



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/workspacePython/function.py (3.6.3) =====
Help on function myFunc in module function:
myFunc()
    내 함수
    리턴은 없음

>>>
```

```
function.py - D:/workspacePython/function.py (3.6.3)
File Edit Format Run Options Window Help
def myFunc(): # """를 활용해 직접 만드는 함수에 도움말 넣기 가능
    """내 함수
    리턴은 없음
    """
    pass

help(myFunc)
```

