

## [Python] 파이썬 기초 9 - 패키지와 모듈에 대한 정의와 다양한 함수 형태 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-23 오후 10:33

URL: <https://continuous-development.tistory.com/69?category=736681>

---

Python

# [Python] 파이썬 기초 9 - 패키지와 모듈에 대한 정의와 다양한 함수 형태

2020. 8. 13. 18:30 수정 삭제 공개



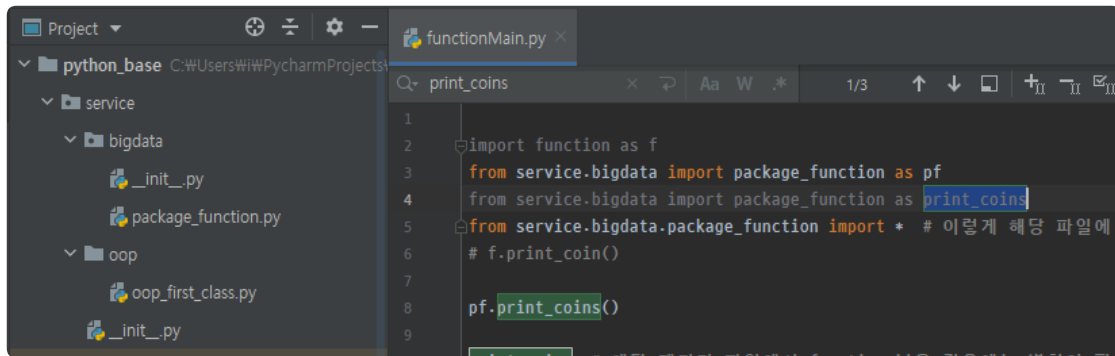
## # 패키지와 모듈

파이썬에서 모듈은 하나의 .py 파일을 가리키며, 패키지는 이러한 모듈들을 모은 컬렉션을 가리킨다. 파이썬의 패키지는 하나의 디렉터리에 놓인 모듈들의 집합을 가리키는데, 그 디렉터리에는 일반적으로 `__init__.py`라는 패키지 초기화 파일이 존재한다.

패키지는 모듈들의 컨테이너로서 패키지 안에는 또다른 서브 패키지를 포함할 수도 있다. 파일 시스템으로 비유하면 패키지는 일반적으로 디렉터

리에 해당하고, 모듈은 디렉터리 안의 파일에 해당한다.

함수 < 클래스 < 모듈 < 패키지



지금 현재 bigdata라는 패키지(폴더)안에 package\_function이라는 모듈이 있다. 이런 일련의 모듈을 합친 게 패키지이다. 그래서 bigdata라는 패키지를 from으로 넣고 그중에 import로 package\_function.py을 했다. functionMain.py이라는 파일은 현재 모듈을 import 해서 해당 모듈에 있는 함수 등을 사용할 수 있다.

```
# package_function.py

def print_coins(): # input 값과 output 이 없는 함수
    for i in range(2):
        print("bitcoin")
```

```
# functionMain.py

import function as f
from service.bigdata import package_function as pf
from service.bigdata import package_function as print_coins
from service.bigdata.package_function import * # 이렇게 해당 파일에 있는 것들을 모두 import 한다.
# f.print_coin()

pf.print_coins() # bitcoin

이런식으로 해당 모듈 pf에 있는 print_coins()를 호출하면 해당 함수가 호출된다.
```

우리가 사용하는 일반 내장함수(sum,max,min 등등)도 위와 같은 경로를 거친다.

## #다양한 형태의 모듈 함수 값

# input 값이(매개변수) 있고 return 값이 없는 함수

```
def first_func(name): # input 값(name) 이 있고 return 값이 없는 함수
    print("Welcone To ~", name)
```

```
pf.first_func("고구마") # Welcome To ~ 고구마
```

# input 값이 있고 return 값도 있는 함수

```
def return_func(name): # input 값이 있고 return 값도 있는 함수
    return "커피 사 줘요~~" + str(name) + "씨"
```

```
# return 값이 있는 형태
thanks = return_func('도라에몽씨')
print(thanks) # 커피 사 줘요~~도라에몽씨
```

# 매개변수를 여러 개 받는 함수

```
def sum_func(x, y, z): # 매개변수를 여러개 받는 함수
    return x + y + z
```

```
# 매개변수가 3개인 형태
sum_value = sum_func(1, 2, 3)
```

```
print("값의 합은 {} 입니다.".format(sum_value)) # 값의 합은 6 입니다.
```

## #tuple 형식으로 매개변수를 받는 함수

```
def tuple_func(*args): # 가변인자로서 매개변수의 개수에 상관없이 다 받는다. 이때 * 이면 가변인.  
    result = 0  
    for idx in range(len(args)):  
        result += args[idx]  
    return result
```

```
# tuple 형태  
result = tuple_func(1, 2, 3, 4, 10)  
print(result)
```

## #dict형식으로 매개변수를 받는 함수

```
def dic_func2(**args): # 가변인자로서 매개변수의 개수에 상관없이 다 받는다. 이때 ** 이면 가변인.  
    for key, value in args.items():  
        print("{} = {}".format(key, value))
```

```
# dic 형태  
dic_func2(name='jslim', name1='park', name2='kim')
```

## #return 값이 여러 개인 형태

```
def multi_func(x, y):  
    sum = x + y  
    mul = x * y  
    return sum, mul # output의 값은 하나인데 튜플형식으로 2개를 보내준다.
```

```
# return 값이 multi 인 형태  
(sum_value, mul_value) = multi_func(3, 4)  
print("합 {} 곱 {}".format(sum_value, mul_value))
```

# 매개변수를 flag로 받는 함수 (생략이 가능하다. 생략일 경우 True로 나타낸다).

```
def default_func(x, y, flag=True): # 매개변수를 안넣는다면 마지막 flag를 따른다. 마지막 매개변수
    sum = x + y
    if sum > 10 and flag:
        return sum
    else:
        return 0
```

```
# flag
result = default_func(10, 20)
print(result)

result = default_func(10, 20, False)
print(result)
```

예제)

```
def count_Sum(start, end):
    oddSum = 0
    evenSum = 0
    for i in range(start, end + 1):
        if i % 2 == 0:
            evenSum += i
        else:
            oddSum += i
    return oddSum, evenSum
```

```
oddSum, evenSum = count_Sum(100, 500)
print("홀수 합, ", oddSum)
print("짝수 합, ", evenSum)

print(calculator(1, '+', 2))
print(calculator(1, '-', 2))
print(calculator(1, '*', 2))
print(calculator(1, '/', 2))
print(calculator(1, 'ㅎ', 2))
```

## 'Python' 카테고리의 다른 글

[Python] 파이썬 기초 11 - 객체의 4대 특성 ( 상속화, 캡슐화, 다형성, 추상화)

[Python] 파이썬 기초 10 - 클래스에 대한 정의와 사용법

**[Python] 파이썬 기초 9 - 패키지와 모듈에 대한 정의와 다양한 함수 형태**

[Python] 파이썬 기초 8 - 반복문(for , while)에 대한 정의와 기본적인 함수 사...

[Python] 파이썬 기초 7 - 조건문(IF, elif ,else)에 대한 정의와 기본적인 함수 사...

[Python] 파이썬 기초 6 - Date type에 대한 정의와 기본적인 함수 사용법

python 모듈

python 패키지

파이썬 모듈

파이썬 패키지



나무늘보스

혼자 끄적끄적하는 블로그 입니다.