
PYTHON
PROGRAMMING
BASIC 2

01

함수

PYTHON

함수 : 입력을 가지고 어떠한 작업을 수행한고, 그 결과를 나타낸다.

왜 사용하는 것일까? :

동일한 작업을 하면서, 이 작업이 다른 작업을 하는데 많이 사용되거나
유의미한 경우에 따로 만들어 두면 여러가지고 편리함.
또한 프로그램의 흐름을 좀 더 쉽게 파악할 수도 있음.

01

함수

PYTHON

일반적인 함수 만들기

```
def 함수이름(입력인수들):
    함수 내에서 수행할 내용 코드들
    return 결과값
```

```
# 더하기 함수
def sum2(a,b):
    result = a + b
    return result

print sum2(10,30)
print 10 + 30
```

```
40
40
```

01

함수

PYTHON

입력이 없는 함수

```
def 함수이름():
    함수 내에서 수행할 내용 코드들
    return 결과값
```

```
def noinput():
    result = "Hello Python!"
    return result

a = noinput()
print a
```

```
Hello Python!
```

01

함수

PYTHON

결과 값이 없는 함수

```
def 함수이름(입력 인수들):
    함수 내에서 수행할 내용 코드들
```

```
def noreturn(a,b):
    temp = a + b
    print "Hello Python!! : " + str(temp)

a = noreturn(2,3)
```

```
Hello Python!! -5
```

01

함수

PYTHON

- 입력/결과 모두 없는 함수

```
def noreturn():
    print "Hello Python!"

noreturn()
```

```
Hello Python!
```

01

함수

PYTHON

입력 값이 여러개가 되는데,
이것이 가변적일 때는?

```
def 함수이름(*입력변수):
    함수 내에서 수행할 내용 코드들
    return 결과값
```

```
def sumrandom(*args):
    result = 0
    for i in args:
        result = result + i
    return result

print sumrandom(1,2,3,4,5,6)
print sumrandom(10,20,30,40,50,60,70,80,90,100)
```

```
21
550
```

01

함수

PYTHON

혼합해서 사용

```
def myfunction(func,*args):
    if func =="sum":
        result = 0
        for i in args:
            result = result + i
        return result
    elif func == "count":
        return len(args)
    else:
        return "No func!"

a = myfunction("sum",10,20,30,40,50)
b = myfunction("count",10,20,30,40,50)
c = myfunction("nono",10,20,30,40,50)
print a
print b
print c
```

```
150
5
No func!
```

01

함수

PYTHON

결과 값을 여러개 사용하려면?

일반적으로 결과값은 1개이나,

여러개를 사용하기 위해서는 파이썬의 튜플형태로 묶어서 1개로 준다.

```
def multifunc(a,b):
    return a+b, a*b, a/b, a+b+2+10
print multifunc(10,30)
```

```
(40,300,0,52)
```

02

함수의 범위

PYTHON

일반적으로 함수 안에서 선언된 변수는 함수 내에서만 사용되고,
함수 외부의 것과 이름이 같아도 다르게 사용된다.

02

함수의 범위

PYTHON

```
a = 1
def func(a)
    a = a + 10
    return a

print func(a)
print(a)
```

```
11
1
```

02

함수의 범위

PYTHON

함수 내부에서 외부의 값을 변경하기(방법1)
-외부 변수에 직접 함수의 결과를 대입해서 사용

```
a = 1
def func(a):
    a = a + 10
    return a

a = func(a)
print(a)
```

```
11
```

02

함수의 범위

PYTHON

- 함수 내부에서 외부의 값을 변경하기(방법2)
  - global 변수 사용

```
a = 1
def func(a):
    global a
    a = a + 10
    return a

print func()
print(a)
```

```
  File"<ipython-input-24-66081895c12d>", line 2
    def func(a):
SyntaxError: name 'a' is local and global
```

```
a = 1
def func():
    global a
    a = a + 10
    return a

print func()
print(a)
```

```
11
11
```

03

lambda 함수

PYTHON

Lambda 함수(익명함수) : 지금까지 배운 함수를 만드는 것과는 달리,
간단하게 한 줄로 필요한 기능을 구현하게 하는 함수.

특징

- 간단하게 만드는 용도기 때문에, 앞에서 일반적인 함수와 달리 중요한 모듈이나 기능을 담당하는 것이 아니라 일회용으로 한 번 정도 사용하는 용도로 사용함.
- 일반적인 함수와 달리 return 값이 없으며, 단지 인자들과 반환값들의 관계식으로 표현됨.
- 특히 데이터 처리를 하는 과정에서는 데이터에 맞는 그 때 그 때 필요한 간단한 수식이나 모듈이 필요한 경우가 많음.
  일반적인 데이터에 해당하는 것이 아니라 특정한 데이터에 국한되는 경우가 많기 때문에 일반적인 함수 대신,
  lambda함수를 사용하는 경우가 많음.
  (예 : 2번째 컬럼의 앞의 2자리 숫자 제거해서 새로운 id 만들기 01_ id001, 02_id002 —〉 id001,id002 등)
- pandas 에서는 map, apply 등과 함께 많이 사용됨! 해당 예제는 추후에 과정에서 여러번 실습할 것이므로,
  여기서는 기본적인 내용만 확인하고 넘어가도록 함.

03

lambda 함수

PYTHON

표현 방법) lambda 인자들 : 표현식

[lambda 함수]

lambda 인자들 : 표현식
lambda 인자1, 인자2, ... : 표현식

```
# a+b의 기존의 함수형식으로 정의하여 사용
def sum2(a,b):
    return a+b
sum2(10,20)
```

30

```
# lambda 형식으로 지정하여 사용하고, 한 번 사용하였기에, 다시 사용하기 위해서는 동일한 코드를 다시 작성해야 함.
(lambda x,y: x+y)(10,20)
```

30

03

lambda 함수

PYTHON

일반적인 함수와 lambda함수의 비교

```
# sum2는 이미 앞에서 정의를 하였기에, 다시 호출해서 사용 가능함.
# lambda는 다시 함수 코드를 작성해야지만 사용이 가능함.
print sum2(20,30)
print (lambda x,y:x+y)(20,30)
```

```
50
50
```

04

과제

PYTHON

- **함수의 입력인자는 2개가 있습니다.**
- **입력인자1 : sel1, sel2** (sel1은 임의의 입력인자 2에 대한 모든 +, * 를 수행한다, sel2 는 입력인자2 중에서 가장 처음 수와 가장 마지막 수를 결과로 준다.)
- **입력인자2 : 입력하는 숫자가 정해지지 않은 숫자들** (단, 2개 이상)
- **입력인자1이 지정한 것 이외의 것이 들어가면 "Check sel parameter!"라는 메시지로 출력하기**

아래의 결과들을 테스트하여 하단의 [결과]와 동일한 결과가 나타나는지 확인하세요.
[테스트]
print myfunc("sel1", 1,2,3,4,5)
print myfunc("sel2", 1,2,3,4)
print myfunc("sel", 1,2,3,4,5)

[결과]
(1,1)
(1,2)
check sel parameter!!!