

# 기초 PYTHON 프로그래밍

## 14. 함수 - 1

1. 함수
2. 파이썬 내장 함수
3. 사용자 정의 함수
4. 함수의 인수와 반환값
5. 함수의 위치와 main 작성하기

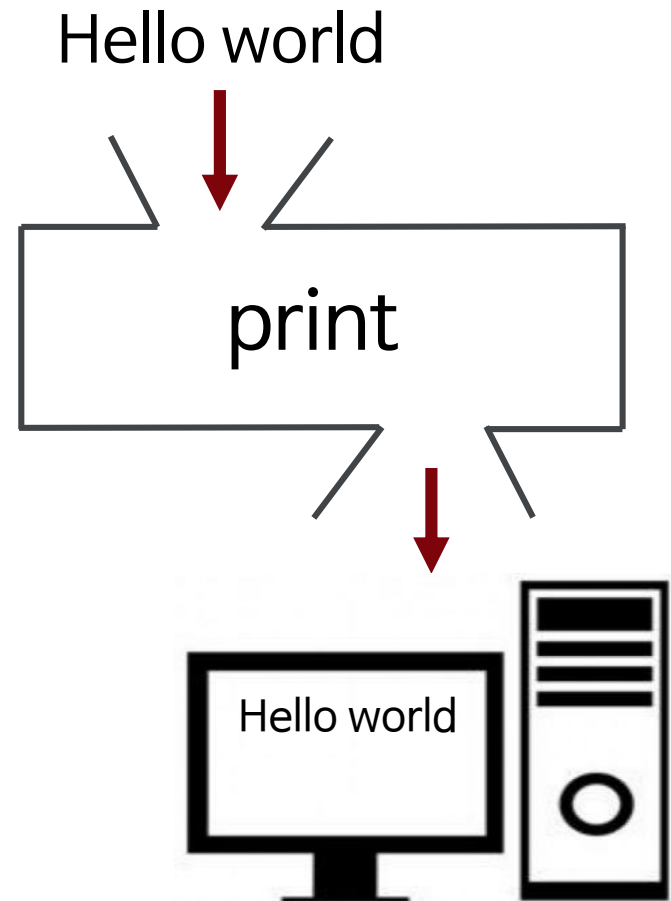
## ◆ 블랙 박스 (black box)

- 함수는 입력과 출력을 갖는 black box이다.
- 주어진 입력에 대해서 어떤 과정을 거쳐 출력이 나오는지 숨겨져 있다.

## ◆ 함수의 재사용성 (reuse)

## ◆ 함수 이름, 입력, 출력 중요함.

```
>>> print('Hello world')
```



## 2. 파이썬 내장 함수

### ◆ 내장 함수 (built-in functions)

- 파이썬 언어에서 미리 만들어서 제공하는 함수들
- IDLE에서 `dir(__builtins__)`라고 입력하면 파이썬에서 제공하는 내장 함수 목록을 볼 수 있다.
- 내장 함수에 어떤 것들이 있는지 학습하고 적절히 사용할 줄 아는 것이 중요하다.

### ◆ 사용자 정의 함수 (user-defined functions)

- 사용자가 직접 만드는 함수
- 함수 작성 문법을 익히고 직접 작성해 보는 것이 중요하다.

## 2. 파이썬 내장 함수

```
>>> dir(__builtins__)  
['ArithmeticError', ..., 'ZeroDivisionError', '_', '__build_class__',  
 '__debug__', '__doc__', '__import__', '__loader__', '__name__',  
 '__package__', '__spec__', 'abs', 'all', 'any', 'ascii', 'bin', 'bool', 'bytearray',  
 'bytes', 'callable', 'chr', 'classmethod', 'compile', 'complex', 'copyright',  
 'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate', 'eval', 'exec', 'exit',  
 'filter', 'float', 'format', 'frozenset', 'getattr', 'globals', 'hasattr', 'hash',  
 'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len',  
 'license', 'list', 'locals', 'map', 'max', 'memoryview', 'min', 'next', 'object',  
 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr',  
 'reversed', 'round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str',  
 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
```

## 2. 파이썬 내장 함수

### ◆ 군집자료형에 유용한 함수들

- len(), max(), min(), sum(), sorted(), reversed() 함수

```
>>> L = [6,4,9,8,10,50,7]
>>> len(L)
7
>>> max(L)
50
>>> min(L)
4
>>> sum(L)
94
>>> sorted(L)
[4, 6, 7, 8, 9, 10, 50]
>>> L
[6, 4, 9, 8, 10, 50, 7]
```

```
>>> T = (4,6,10,2,1,5)
>>> a = len(T)
>>> b = max(T)
>>> c = min(T)
>>> d = sum(T)
>>> print(a,b,c,d)
6 10 1 28

>>> S = sorted(T)
>>> print(S)
[1, 2, 4, 5, 6, 10]
>>> type(S)
<class 'list'>
```

```
>>> S = {4,6,2,7,5}
>>> a = len(S)
>>> b = max(S)
>>> c = min(S)
>>> d = sum(S)
>>> print(a,b,c,d)
5 7 2 24

>>> T = sorted(S)
>>> print(T)
[2, 4, 5, 6, 7]
>>> type(T)
<class 'list'>
```

## 2. 파이썬 내장 함수

### ◆ reversed 함수 - 군집자료형 데이터를 역순으로 바꾼다.

- 순서가 있는 list, tuple, str 에만 적용할 수 있다.

```
>>> L = [1,3,5,7,9]
>>> M = reversed(L)
>>> print(M)
<list_reverseiterator object at 0x03DC92B0>
>>> M = list(reversed(L))
>>> print(M)
[9, 7, 5, 3, 1]
```

```
>>> T = (2,4,6,8)
>>> K = list(reversed(T))
>>> print(K)
[8, 6, 4, 2]
```

```
>>> S = 'HELLO'
>>> P = list(reversed(S))
>>> print(P)
['O', 'L', 'L', 'E', 'H']
```

### 3. 사용자 정의 함수

#### ◆ 함수 정의하기

```
def 함수명 ( parameter ) :  
    """ 이 함수는 두 정수에서 큰 값  
    을 찾아서 반환하는 함수이다. """  
  
    return 반환값
```

parameter  
(매개변수)

**a**      **b**

```
if a > b :  
    y = a  
else :  
    y = b
```



**y**

반환값

함수명 : find\_max

```
def find_max (a,b):  
    if a > b :  
        y = a  
    else :  
        y = b  
    return y
```

◆ docstring : """ ... .. """

◆ 함수명 - 변수명 규칙과 같다.

- 영문 대소문자, 숫자, \_ 로 구성됨.
- 숫자로 시작할 수 없음.

### 3. 사용자 정의 함수

- ◆ 함수 호출하기 - 함수명에 인수를 넘기면서 호출함.

```
def find_max (a,b):  
    if a > b :  
        y = a  
    else :  
        y = b  
    return y
```

**a,b** : 매개변수 (parameter)

# main

```
m = find_max(10,20) # 10, 20 : 인수 (arguments)  
n = find_max(7,3)   # 7, 3 : 인수 (arguments)  
print(m,n)
```



## 4. 함수의 인수와 반환값

### ◆ 인수, 반환값이 없는 함수

- 함수에 입력이 없으면 빈 괄호로 둔다.
- 함수에 출력이 없을 수도 있다 (return 구문이 없다).

```
def hello():  
    print('hello world')  
    print('hello python~')  
  
# main - 여기에서 프로그램 수행 시작  
  
print('start of the program')  
hello()  
print('middle of the program')  
hello()  
print('end of the program')
```

```
start of the program  
hello world  
hello python~  
middle of the program  
hello world  
hello python~  
end of the program
```

## 4. 함수의 인수와 반환값

### ◆ 인수가 여러 개인 함수 (인수 개수만큼 parameter 필요함)

- 예) main 에서 국어, 영어, 수학 성적을 입력받아 calculate함수의 입력으로 넘긴다. calculate 함수는 평균을 반환한다.

```
def calculate(kor, eng, math):  
    total = kor + eng + math  
    average = total / 3  
    return average
```

```
국어 성적을 입력하시오 : 90  
영어 성적을 입력하시오 : 85  
수학 성적을 입력하시오 : 92  
평균 : 89.0
```

#### # main

```
kor_score = int(input('국어 성적을 입력하시오 :'))  
eng_score = int(input('영어 성적을 입력하시오 :'))  
math_score = int(input('수학 성적을 입력하시오 :'))  
  
avg = calculate(kor_score, eng_score, math_score)  
print('평균 : ', avg)
```

## 4. 함수의 인수와 반환값

- ◆ 함수에서 return 문이 여러 번 나오는 경우
  - return 문이 여러 번 나오더라도 가장 먼저 return을 만나는 순간 함수는 값을 반환하고 종료한다.

```
def find_max (a,b):  
    if a > b:  
        y = a  
    else:  
        y = b  
    return y
```

함수명 : find\_max

```
def find_max (a,b):  
    if a > b:  
        return a  
    else:  
        return b
```

## 4. 함수의 인수와 반환값

### ◆ 반환값이 여러 개인 경우

- 함수에 반환값이 2개 이상인 경우 **튜플**로 묶어서 반환한다.
- 예) 두 수를 입력받아서 두 수의 합과 두 수의 곱을 반환하는 함수

```
def add_multiply(x,y):  
    sum = x + y  
    mult = x * y  
    return sum, mult # 튜플로 반환한다.
```

```
# main  
a = int(input('Enter a : '))  
b = int(input('Enter b : '))  
m,n = add_multiply(a,b)  
print(m,n)
```

## 5. 함수의 위치와 main 작성하기

### ◆ 함수의 위치

- 함수는 호출 전에 정의되어 있어야 한다.

```
print('start of the program')
hello()
print('middle of the program')
hello()
print('end of the program')

def hello():
    print('hello world')
```

```
start of the program
Traceback (most recent call last):
  File "C:/Users/sogang-pc/AppData/Local
ne 2, in <module>
    hello()
NameError: name 'hello' is not defined
```

```
a = 10
b = 20
print('before test definition')
```

```
def test():
    print('I am test')
    print(a, b)
```

```
print('start')
test()
print('end')
```

```
before test definition
start
I am test
10 20
end
```

## 5. 함수의 main 작성하기

### ◆ main 함수 작성하기

- main은 프로그램 시작점을 의미한다.

```
def find_max (a,b):  
    if a > b: return a  
    else: return b
```

```
x = find_max(10, 20)  
y = find_max(7,3)  
print(x,y)
```

```
def find_max (a,b):  
    if a > b: return a  
    else: return b
```

```
def main():  
    x = find_max(10, 20)  
    y = find_max(7,3)  
    print(x,y)
```

```
if __name__ == '__main__':  
    main()
```