함수의 이해

8주차_01

한 동 대 학 교
김경미 교수

함수사용 이유

- **코드가 길어질 때 모듈화하여 간결성 높인다**
- **반복되는 구분을 모듈화 하여**
  - 고치기 쉽고(easy to modify)
  - 운영과 관리를 용이하게 하며(flexible to maintain)
  - 프로그램 가독성을 높게 한다(readability)

함수

- **함수의 종류**
  - 내장 함수(Built-in function)
  - 사용자 정의 함수(User defined function)

- **함수를 정의하려면**
  - 함수 이름과 명령문들을 순서대로 쓴다
  - 함수 이름을 불러서 함수를 “호출” 한다 (function call)

사용해 본 내장함수

- **print()**
- **input()**
- **int()**
- **float()**
- **range()**
- **len()**
- **deepcopy()**

내장 함수(Built-in Function)

- **시스템에서 제공해 주는 함수들**
  - 사용한 내장 함수

```
>>> int('55')
>>> print( 'kmkim' )
>>> range(10)
```

  - 어떤 함수들은 모듈을 import한 후 사용 가능

```
>>> import math
>>> math.sqrt(4) / 2.0
1.0
```

내장함수 활용 1

```
# type conversion function
>>> int('32')
32
>>> int(3.99999)
3
>>> float(-2)
-2.0
#  Unicode -> character
>>> chr(65)
'A'
>>> chr(97)
'a'

#  character -> unicode
>>> ord('a')
97
>>> ord("F")
70
```

내장함수 활용 2

```
>>> print(5)
5
>>> i = input("enter your age; ")
enter your age; 12
>>> print(i * 2)
1212

# sum()
>>> num_list=[1,2,3,4,5]
>>> sum(num_list)
15

#sorted()
>>> num_l=[5,6,2,9,1]
>>> sorted(num_l)
[1, 2, 5, 6, 9]
>>> num_l
[5, 6, 2, 9, 1]
```

내장함수 활용 3

http://docs.python.org/3.3/library/functions.html

abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

내장함수 활용 4

- **파이썬에는 친숙한 수학적 함수들을 제공하는 math module이 있다**
  - 모듈은 관련된 함수들의 모음이다
  - 모듈을 사용하려면 import 사용해야 한다

```
# math module
>>> import math
>>> print(math)
<module 'math' (built-in)>

>>> degrees = 45
>>> radians = degrees / 360.0 * 2 * math.pi
>>> math.sin(radians)
0.707106781187
>>> math.sqrt(2) / 2.0
0.707106781187
```

내장함수, Argument and return value

```
>> int_num = int(23.5)
>> int_num
23
```

argument

return value

```
>> r = round(2.3456, 2)
```

2 arguments

Function call

- Argument는 함수를 호출했을 때 값을 받게 된다
- 리턴 값은 함수가 끝날 때 그 결과를 기억한다

내장함수, Sorting

- **정렬**
  - 특정한 순차(sequence) 혹은 다른 집합(set)들에 따라 항목(item)을 배열하는 과정을 말한다
  - 순서화(Ordering)
    - 같은 종류, 클래스의 항목(items)들을 순서 있는 나열로 배열하는 것
  - 카테고리화(Categorizing)
    - 비슷한 특징을 지닌 항목끼리 그룹화(grouping)하고 표시하는 것

내장함수, Sorting

https://wiki.python.org/moin/HowTo/Sorting/

```
>>> a = [5, 2, 3, 1, 4]
>>> a.sort()
>>> a
[1, 2, 3, 4, 5]


>>> data = [('red', 1), ('blue', 1), ('red', 2), ('blue', 2)]
>>> sorted(data, key=itemgetter(1))                    # sort by second item
[('blue', 1), ('red', 1), ('blue', 2), ('red', 2)]
>>> data
[('red', 1), ('blue', 1), ('red', 2), ('blue', 2)]
```

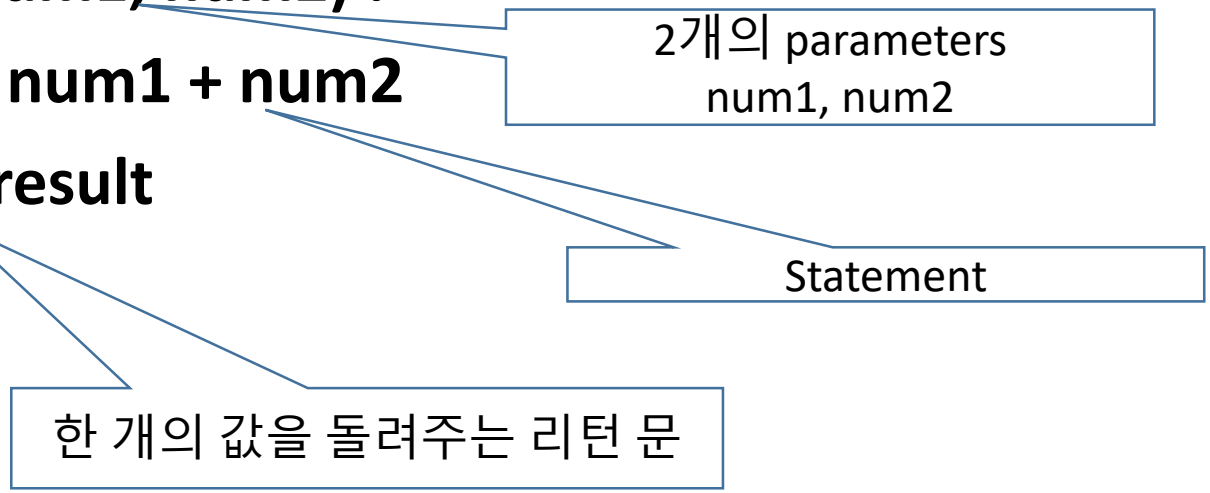
사용자 정의 함수(User defined function)

- **사용자 정의 함수란?**
  - 사용자가 필요하다고 판단하는 루틴을 구상하여 함수로 정의하고 사용하는 것

- **함수 정의하기(define function)**
- **선언된 함수 사용하기(function call)**

함수 정의하기

```
def  plus(num1, num2) :
    result = num1 + num2
    return  result
```



사용자 정의 함수 정의 과정

- 1단계. 함수의 기능을 정리하고, 이름을 정한다
- 2단계. 함수에서 입력과 결과가 무엇인지 정한다
- 3단계. 입력으로 사용하는 파라미터와 리턴값의 데이터형을 정한다
- 4단계. 원하는 함수의 기능을 pseudocode로 쓴다
- 5단계. 함수를 만든다
- 6단계. 제대로 실행되는지 테스트 한다

사용자 정의함수 만들기 예제

- **3개 숫자를 입력 받아서 평균을 계산하여 돌려준다**
  - 평균을 구하는 함수 이름은 calculate_avg

- **Parameter는 3개, 데이터형 : int와 float → float**
  return value 데이터형: float

- **Pseudocode 써 보기**

  sum = parameter1 + parameter2 + parameter3

  return sum/3

사용자 정의함수 만들기 예제

- 함수 만들기

```
def calculate_avg(n1, n2, n3):
    sum = n1 + n2 + n3
    return sum/3
```

- 제대로 실행되는지 확인해 본다

사용자 정의함수 만들기 예제

- **정의한 함수 사용해 보기**

```
# Define a function
def calculate_avg(n1, n2, n3):
      sum = n1 + n2 + n3
      return sum/3

# Call the function
avg=calculate_avg(1,5,9)
print("result of function call = ", avg)

avg=calculate_avg(11, 9, 16)
print("result of function call = ", avg)
```

```
>>>
======= RESTART: E:/1_Works/2017Work.
======
result of function call =  5.0
result of function call =  12.0
>>>
```

요약

- 함수를 왜 사용하는지 이해한다
- 내장함수가 무엇인지 이해한다
- 사용자 정의 함수를 만드는 과정을 따라해 본다

감사합니다

8주차_01 함수의 이해