

파이썬과 함께하는 이세계 코딩 생활

5주차 : 함수와 리스트



오늘 배울 내용

- 함수
- 리스트

함수 (function)

- 함수란 무엇인가?
 - : 특정 목적의 작업을 수행하기 위해 독립적으로 구성된 프로그램 코드의 집합
- 함수가 왜 필요한가?
 1. 코드의 불필요한 반복을 피하기 위해서
 2. 프로그램의 가독성을 위해서
 3. 함수 내에서 독립적인 변수를 선언하여 사용할 수 있게 하기 위해서
→ 동일한 변수 명을 다른 함수에서 사용할 수 있다.
 4. 프로그램을 여러 부분으로 나누어 프로그래밍 && 컴파일 하기 위해서

function ?

```
def 함수명 (매개변수) :  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
    return 결과값
```

함수에는
3가지 종류가
있다!

1. 빌트-인 함수
2. 모듈 함수
3. 사용자 정의 함수

빌트-인 함수

```
help(print)
```

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 stream, or to sys.stdout by default.

Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

end: string appended after the last value, default a newline.

flush: whether to forcibly flush the stream.

이 함수는 Return 값이 없다. 따라서 None 을 리턴한다. (다른 언어에서의 NULL 과 동일)



빌트-인 함수

```
help(sum)
```

Help on built-in function sum in module **builtins**:

```
sum(iterable, start=0, /)
```

Return the sum of a 'start' value (default: 0) plus an iterable of numbers

When the iterable is empty, return the start value.

This function is intended specifically for use with numeric values and may reject non-numeric types.

이 함수는 Return 값이 있다. 설명을 읽어보면 숫자 형태의 특정 값을 리턴한다는 걸 알 수 있다.



모듈 함수

```
import random # 꼭 해줘야함!  
help(random.randint)
```

Help on method randint in module **random**:

randint(a, b) method of random.Random instance

Return random integer in range [a, b], including both end points.

이 함수는 정수(integer)를 리턴한다.

[a, b] : a 이상 b 이하

[: 이상

] : 이하

(: 초과

) : 미만

사용자 정의 함수

```
def sum_digit(n):  
    result = 0  
    while n > 0:  
        result += n%10  
        n = int(n/10)  
    return result
```

```
help(sum_digit)
```

```
Help on function sum_digit in module __main__:  
sum_digit(n)
```

설명이 없어서 뭐하는 함수인지 잘 모르겠어요! 😊



사용자 정의 함수 (+Docstring)

```
def sum_digit(n):  
    '''  
    정수 n의 각 자릿수를 더해서 리턴한다.  
    '''  
    result = 0  
    while n > 0:  
        result += n%10  
        n = int(n/10)  
    return result
```

```
help(sum_digit)
```

```
Help on function sum_digit in module __main__:  
sum_digit(n)  
    정수 n의 각 자릿수를 더해서 리턴한다.
```

Docstring('\" , \"\"\"\" 사용) 을 이용해서 문서화

- one-line, multi-line 모두 가능

Docstring Conventions

<https://www.python.org/dev/peps/pep-0257/>

💡 함수에서 매개변수는 있어도 되고 없어도 되나요? (네!)

매개변수가 없는 함수

```
def print_hi():  
    print('hi')
```

```
print_hi()
```

```
hi
```

매개변수가 있는 함수

```
def print_hi_name(name):  
    print('hi ' + name)
```

```
name = 'interface'  
print_hi_name(name)
```

```
hi interface
```

💡 함수에서 리턴 값은 있어도 되고 없어도 되나요? (네!)

Return 값이 없는 함수는 None 을 리턴한다. (다른 언어에서의 NULL 과 동일)

```
a = print('hi')  
print(a)
```

```
hi  
None
```

Return 값이 있는 함수는 특정 값을 리턴한다. (ex. Integer, float, string ... 형태)

```
b = sum([1, 2, 3])  
print(b)
```

아까 sum 함수에서 [] 안에 값이 들어간 형태가 있었다.
대충 보면 sum 함수는 이 안에 들어 있는 값들을 모두 더해서
리턴한 거 같긴 한데... 이게 뭘까? 😊

리스트 (List)

- 리스트란 무엇인가?
 - : 원소들이 연속적으로 저장되는 형태의 자료형
- 특징
 - 리스트는 대괄호([,])로 감싸서 나타낸다. (ex. [1,2,3])
 - 리스트 안에는 0개 이상의 원소가 저장될 수 있다.
 - 서로 다른 자료형의 원소도 저장될 수 있다. (ex. ['hi', 5, 1.234])
 - 리스트 안에 원소로 리스트를 저장할 수도 있다. (ex. 2차원 리스트 : [[1,2,3], [4,5,6]])

예시1

첫째 줄에 과목 갯수 n

두번째 줄부터 n 줄에 걸쳐서 점수 n 개 입력 받음

리스트에서 하나하나 비교해서 정렬

최대값, 최소값, 평균값(소수점 둘째 자리까지) 구해서 출력

예시1 : 코드

```
n = int(input())
score_list = []
sum_score = 0

for i in range(n):
    score = int(input())
    score_list.append(score)
    sum_score += score

# 정렬
for i in range(n):
    for j in range(i+1, n):
        if score_list[i] > score_list[j]:
            tmp = score_list[i]
            score_list[i] = score_list[j]
            score_list[j] = tmp

# print(score_list)

# 결과
print('max = %d' % score_list[n-1]) # print('max = %d' % score_list[-1])
print('min = %d' % score_list[0])
print('avg = %.2f' % (sum_score*(1.0)/n))
```



예시2 (sort() 사용)

첫째 줄에 과목 갯수 n

두번째 줄부터 n 줄에 걸쳐서 점수 n 개 입력 받음

리스트 함수 이용해서 정렬

최대값, 최소값, 평균값(소수점 둘째 자리까지) 구해서 출력

예시2 : 코드

```
n = int(input())

score_list = []

sum_score = 0

for i in range(n):
    score = int(input())
    score_list.append(score)
    sum_score += score

# 정렬
score_list.sort()

# print(score_list)

# 결과
print('max = %d' % score_list[n-1]) # print('max = %d' % score_list[-1])
print('min = %d' % score_list[0])
print('avg = %.2f' % (sum_score*(1.0)/n))
```



예시3 (max(), min(), avg() 사용)

첫째 줄에 과목 갯수 n

두번째 줄부터 n 줄에 걸쳐서 점수 n개 입력 받음

내장 함수 이용해서 최대값, 최소값, 평균값(소수점 둘째 자리까지) 출력

예시3 : 코드

```
n = int(input())
score_list = []

for i in range(n):
    score = int(input())
    score_list.append(score)

# 결과
print('max = %d' % max(score_list))
# print('max = %d' % score_list[-1])
print('min = %d' % min(score_list))
print('avg = %.2f' % (sum(score_list)/(len(score_list)))) #score_list 그냥 n 써도 됨! (len이란게 있다~)
```



예시4 (한 줄로 입력, 함수로 만들기)

첫째 줄에 점수 n 개 입력 받음

리스트 함수 이용해서 최대값, 최소값, 평균값(소수점 둘째 자리까지) 출력

이걸 `print_things` 라는 이름의 함수로 만들어라!

예시3 : 코드

```
def my_function():  
    score_list = list(map(int, input().split()))  
    print(score_list)  
    print('max = %d' % max(score_list))  
    print('min = %d' % min(score_list))  
    print('avg = %.2f' % (sum(score_list)/(len(score_list))))  
#score_list 그냥 n 써도 됨! (len이란게 있다~)  
  
my_function()
```



리스트 관련 함수

- append()
- sort()
- reverse()
- index()
- insert()
- remove()
- pop()
- count()
- extend()

리스트 관련 함수

```
help(list.append)
```

```
Help on method_descriptor:
```

```
append(self, object, /)
```

```
    Append object to the end of the list.
```

리스트 관련 함수

```
help(list.sort)
```

```
Help on method_descriptor:
```

```
sort(self, /, *, key=None, reverse=False)
```

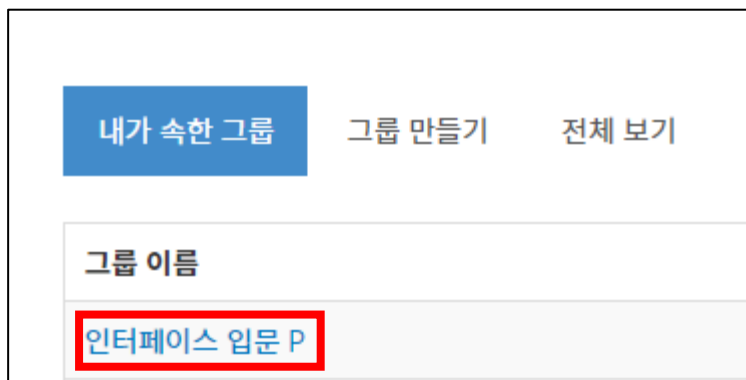
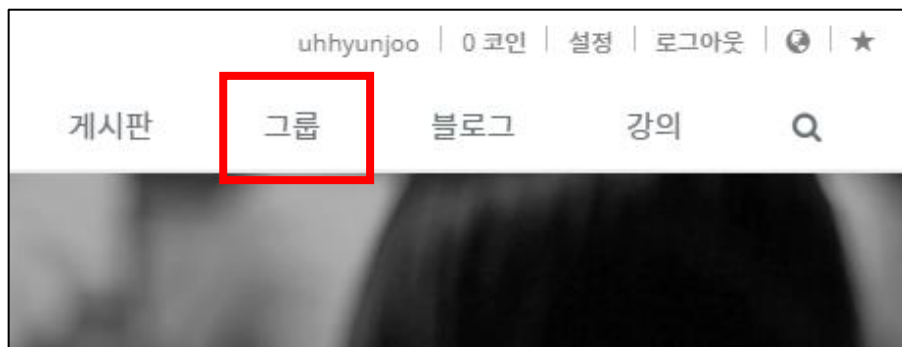
```
    Stable sort *IN PLACE*.
```

이런 식으로 help 를 이용해서 보면 무슨 의미인지 알 수 있다~
다른 함수들도 한 번씩 해보도록 하여!



백준으로 파이썬 맛보기

- 5주차 문제집



문제 번호	제목	정보	맞은 사람	제출	
10818	3 최소, 최대	성공 분류	37916	110303	43.763%
2562	2 최댓값	성공 출처 분류	35007	82451	48.839%
2577	2 숫자의 개수	성공 출처 분류	41882	82704	60.529%
3052	2 나머지	성공 출처 다국어 분류	29158	55377	60.898%
1546	1 평균	성공 스페셜 저지 분류	37867	91921	48.924%
8958	2 OX퀴즈	성공 출처 다국어 분류	36274	84536	51.540%
4344	1 평균은 넘겠지	성공 출처 다국어 분류	34595	110806	37.318%

