

(실습) 함수 정의 및 실행

백석대학교 강윤희

(프로그램) 함수 정의 및 실행

- 두 숫자를 받아 큰 수와 작은 수를 인쇄하는 함수 프로그램

```
In [1]: def max(x,y):  
        if x > y:  
            return x  
        else:  
            return y
```

```
In [2]: def min(x,y):  
        if x < y:  
            return x  
        else:  
            return y
```

```
In [3]: print ("첫번째 숫자를 입력하세요")  
        x = input()
```

첫번째 숫자를 입력하세요
5

```
In [4]: print ("두번째 숫자를 입력하세요")  
        y = input()
```

두번째 숫자를 입력하세요
10

```
In [5]: print ("두수의 큰값은 찾으시오 = %d" % max(x,y))
```

```
In [6]: x = int(x)  
        y = int(y)  
        print ("두수의 큰값은 찾으시오 = %d" % max(x,y))
```

두수의 큰값은 찾으시오 = 10

(프로그램) 함수 정의 및 실행

- 양의 정수 하나를 받아 1부터 전달받은 정수까지의 합을 구하여 반환하는 함수 프로그램

```
In [1]: def sum_to(x):  
        sum = 0  
        for i in range(1, x+1):  
            sum = sum + i  
        return sum
```

```
In [12]: print("1에서 합을 구하고자 하는 숫자를 입력하세요. ")  
x = input ()  
x = int(x)  
print("{} 까지 합".format(x))  
print(sum_to(x))
```

(프로그램) 함수 정의 및 실행

- 키보드를 통하여 학생 점수를 0점 숫자가 나올 때까지 입력 받은 후 0점을 제외한 나머지 점수의 평균을 리턴하는 함수 만들기

```
In [13]: def get_average():  
        scores = []  
        while True:  
            print ("input number :")  
            score = int(input())  
            if score != 0:  
                scores.append(score)  
            else:  
                break  
        # print (scores)  
        return sum(scores)/len(scores)  
  
        avg = get_average()  
        print ("average = %f" % avg)
```

```
input number :  
10  
input number :  
20  
input number :  
30  
input number :  
0  
average = 20.000000
```

(프로그램) 함수 정의 및 실행

- 숫자 리스트 nums 를 전달받아 리스트에 0이 있는지를 판별하는 함수 test 만들기

```
In [5]: def test(nums):  
        for i in nums:  
            if i == 0: # if the condition is satisfied, it hits break and the else block will not run  
                print('There is a 0.')  
                break  
            else:  
                print('There are no 0s.')
```

```
In [6]: test([1,2,3,0]) # There is a 0.  
        test([1,2,3])   # There are no 0s.
```

```
There are no 0s.  
There are no 0s.  
There are no 0s.  
There is a 0.  
There are no 0s.  
There are no 0s.  
There are no 0s.
```

반복적으로 출력되는 것을 없애고자 함

for문 바깥쪽에 else문을 추가해서, break가 발생하지 않았을 때의 동작에 대해 기술
즉, for ~ else문은 "for문에서 break가 발생하지 않았을 경우" 의 동작을 else문에 적어주는 것임

(프로그램) 함수 정의 및 실행

- 숫자를 갖는 리스트 nums 를 전달받아 리스트에 0이 있는지를 판별하는 함수 test 만들기

```
In [1]: def test(nums):  
        for i in nums:  
            if i == 0: # if the condition is satisfied, it hits break and the else block will not run  
                print('There is a 0.')  
                break  
        else:  
            print('There are no 0s.')
```

```
In [2]: test([1,2,3,0]) # There is a 0.  
        test([1,2,3])   # There are no 0s.
```

There is a 0.
There are no 0s.

- 참고: for – else 활용