

Ch 9-A: Python Function Practice with Sophistication

`map()`, `filter()`, Lambda function, `sorted()`, `reversed()`, `max()`,
`enumerate()`, `zip()` 등등을 이용하여 간결한 code를 만들도록
노력해 본다!

1. 지정된 리스트의 홀수 요소 수를 반환하는 함수 f1(list)을 작성하세요.

```
>>> f1([1,2,3,4])
```

```
2
```

```
>>> f1([1,2,3,4,5])
```

```
3
```

```
def f1(li):  
    num = 0  
    for i in li:  
        if i % 2 == 1:  
            num = num + 1  
    return num
```

2. 주어진 리스트에서 각 홀수 요소를 출력하는 함수 f2(list)를 작성하세요.

```
>>>f2([1,2,3,4])
```

```
1
```

```
3
```

```
>>>f2([1,2,3,4,5])
```

```
1
```

```
3
```

```
5
```

```
def f2(list):  
    for i in range( 0, len(list) ) :  
        if list[i] % 2 == 1 :  
            print ( list[i] )
```

3. 주어진 리스트에 있는 모든 홀수 요소의 합계를 반환하는 함수 f3(list)을 작성하세요.

```
>>> f3([1,2,3,4])
```

```
4
```

```
>>> f3([1,2,3,4,5])
```

```
9
```

```
def f3(list):  
    sum = 0  
    for i in range( 0, len(list) ) :  
        if list[i] %2 == 1 :  
            sum = sum + list[i]  
    return sum
```

4. 지정된 리스트에서 요소가 홀수에 해당하는 모든 인덱스의 합계를 반환하는 함수 `f4(list)`를 작성하세요.

```
>>> f4([1,2,3,4])
```

```
2
```

```
>>> f4([1,2,3,4,5])
```

```
6
```

```
def f4(list) :  
    sum = 0  
    for i in range( 0, len(list) ) :  
        if list[i] % 2 == 1 :  
            sum = sum + i  
    return sum
```

5. 각 요소가 제공된 동일한 리스트를 반환하는 함수 f5(list)를 작성하세요.

```
>>> f5([1,2,3,4])  
[1,4,9,16]  
>>> f5([1,2,3,4,5])  
[1,4,9,16,26]
```

```
def f5(list) :  
    for i in range( 0, len(list) ):  
        list[i] = list[i] ** 2  
    return list
```

6. 지정된 리스트에서 가장 큰 숫자를 반환하는 함수 f6(list)를 작성하세요.

```
>>> f6([1,2,3,4])
```

```
4
```

```
>>> f6([1,2,3,4,5])
```

```
5
```

```
def f6(list) :  
    largest = list[0]  
    for i in range( 0, len(list) ):  
        if largest < list[i] :  
            largest = list[i]  
    return largest
```

7. 주어진 리스트에 있는 모든 숫자의 평균을 반환하는 함수 f7(list)을 작성하세요.

```
>>> f7([1,2,3,4])
```

```
2.5
```

```
>>> f7([1,2,3,4,5])
```

```
3.0
```

```
def f7(list):  
    sum = 0  
    for i in range(0, len(list)):  
        sum = sum + list[i]  
    return sum / len(list)
```


8. a와 b 포함 범위 내에서 n으로 나누어 떨어질 수 있는 모든 숫자를 출력하는 함수 f8(a,b,n)을 작성하세요.

n은 양수라고 가정합니다.

```
>>> f8(1,10,2)
```

```
2
```

```
4
```

```
6
```

```
8
```

```
10
```

```
>>> f8(1,10,11)
```

```
>>> f8(1,10,7)
```

```
7
```

```
def f8(a,b,n):  
    for i in range( a, b+1 ):  
        if i % n == 0:  
            print ( i )
```

9. 주어진 너비와 높이로 ASCII 직사각형을 출력하는 함수 f9(width,height)를 작성하세요.

```
>>> f9(0,1)
>>> f9(10,0)
>>> f9(1,1)
*
>>> f9(1,2)
*
*
>>> f9(5,5)
*****
*****
*****
*****
*****
```

```
def f9(width, height):
    for i in range(0, height):
        for j in range(0, width):
            print(" ", end=" ")
        print()
```

10. 주어진 높이 n 으로 삼각형을 출력하는 함수 $f10(n)$ 을 작성하세요. n 이 음수가 아니라고 가정합니다.

```
>>> f10(1)
```

```
*
```

```
>>> f10(2)
```

```
*
```

```
**
```

```
>>> f10(3)
```

```
*
```

```
**
```

```
***
```

```
def f10(n):  
    for i in range(0, n):  
        for j in range(0, i+1):  
            print(" ", end="")  
        print()
```

11. 리스트가 내림차순으로 정렬되어 있으면 True를 반환하고 그렇지 않으면 False를 반환하는 함수 f11(list)을 작성하세요. 빈 리스트는 True를 반환합니다.

```
>>> f11([])
True
>>> f11([5,4,3,2,1])
True
>>> f11([5,4,3,2,0])
True
>>> f11([5,4,5,2])
False
```

```
def f11(list):
    for i in range( 0 , len(list)-1 ) :
        if list[i] < list[i+1] :
            return False
    return True
```

12. 리스트가 모든 음수로 구성된 경우 True를 그렇지 않으면 False를 반환하는 함수 f12(list)를 작성하세요.

빈 리스트는 True를 반환합니다.

```
>>> f12([])
```

```
True
```

```
>>> f12([-1,-2,-3,-4,5])
```

```
False
```

```
>>> f12([1,2,3,4,5])
```

```
False
```

```
>>> f12([-1,-2,-3])
```

```
True
```

```
def f12(list):  
    for i in range( 0, len(list) ) :  
        if list[i] > 0 :  
            return False  
    return True
```

13. 리스트에서 마지막으로 존재하는 target의 인덱스를 반환하는 함수 f13(list, target)을 작성하세요.
리스트가 비어 있지 않고 항상 target이 포함된다고 가정합니다.

```
>>> f13([1,2,3], 3)
2
>>> f13([1,2,3,1,2,3], 3)
5
>>> f13([1,1,1,1], 1)
3
```

```
def f13(list, target) :
    last_index = 0
    for i in range ( 0, len(list) ) :
        if list[i] == target :
            last_index = i
    return last_index
```

14. 리스트의 마지막 음수 인덱스를 반환하는 함수 f14(list)를 작성하세요.
리스트가 비어 있지 않고 항상 음수가 포함된다고 가정합니다.

```
>>> f14([1,2,-3])
2
>>> f14([1,-2,-3,1,-2,-3])
5
>>> f14([-1,1,1,1])
0
```

```
def f14(list) :
    last_index = 0
    for i in range( 0, len(list) ) :
        if list[i] < 0 :
            last_index = i
    return last_index
```

15. 짝수 인덱스의 모든 요소의 합계를 반환하는 함수 f15(list)를 작성합니다.

```
>>> f15([1,2,-3])  
-2  
>>> f15([1,-2,-3,1,-2,-3])  
-4  
>>> f15([-1,1,1,1])  
0
```

```
def f15(list) :  
    i = 0  
    sum = 0  
    while i < len(list) :  
        sum = sum + list[i]  
        i = i+2  
    return sum
```


16. 거꾸로 된 삼각형을 출력하는 함수 f16(n)을 작성하세요.

```
>>> f16(3)
```

```
***
```

```
**
```

```
*
```

```
>>> f16(2)
```

```
**
```

```
*
```

```
>>> f16(1)
```

```
*
```

```
def f16(n):  
    for i in range(0,n):  
        for j in range( n-i ) :  
            print ( "*", end="" )  
        print ()
```

17. 리스트의 요소들을 역순으로 2개 건너씩 출력하는 함수 f17(list)을 작성하세요.

```
>>> f17([1,2,3,4,5,6])
6
4
2
>>> f17([1,2,3,4])
4
2
>>> f17([1])
1
```

```
def f17(list):
    i = len(list)-1
    while i >= 0 :
        print ( list[i] )
        i = i-2
```

18. $n!$ 을 반환하는 함수 $f18(n)$ 을 작성하세요.

```
>>> f18(n)
```

```
1
```

```
>>> f18(2)
```

```
2
```

```
>>> f18(3)
```

```
6
```

```
def f18(n):  
    factorial = 1  
    for i in range( 1, n+1 ) :  
        factorial = factorial * i  
    return factorial
```

19. 매트릭스의 각 행의 합계를 출력하는 함수 f19(matrix)를 작성하세요.

```
>>> f19([[1,0],[0,1]])
1
1
>>> f19([[1,2,3],[4,5,6]])
6
15
>>> f19([[1],[2],[3],[4]])
1
2
3
4
```

```
def f19(matrix):
    for row in range( 0, len(matrix) ) :
        sum = 0
        for col in range( 0, len(matrix[row])):
            sum = sum + matrix[row][col]
        print ( sum )
```

20. 매트릭스의 대각선을 출력하는 함수 f20(matrix)을 작성하세요.
행렬이 정사각행렬이라고 가정합니다.

```
>>> f20([[1,0],[0,1]])
```

```
1
```

```
1
```

```
>>> f20([[1,2,3],[4,5,6],[7,8,9]])
```

```
1
```

```
5
```

```
9
```

```
>>> f20([[1]])
```

```
1
```

```
def f20(matrix):  
    for row in range(0, len(matrix)):  
        for col in range(0, len(matrix[row])):  
            if row == col:  
                print (matrix[row][col])
```

21. 주어진 목록의 각 요소의 팩토리얼을 출력할 함수 f21(list)을 작성하세요.

```
>>> f21([])
>>> f21([1,2,3])
1
2
6
>>> f21([1,2,3,4])
1
2
6
24
```

```
def f21(list):
    for i in range( 0, len(list) ):
        factorial = 1
        for j in range( 1, list[i]+1 ):
            factorial = factorial*j
        print ( factorial )
```

22. 주어진 리스트에 대해 각 요소에서 0으로 시작하는 카운트다운을 출력하는 함수 f22(list)을 작성하세요.

```
>>> f22([])
>>> f22([1,3,5])
1 0
3 2 1 0
5 4 3 2 1 0
>>> f22([5,3,6,2])
5 4 3 2 1 0
3 2 1 0
6 5 4 3 2 1 0
2 1 0
```

```
def f22(list):
    for i in range(0, len(list)):
        countdown = list[i]
        while countdown >= 0:
            print (countdown, end=" ")
            countdown = countdown - 1
        print ()
```

23. 새 리스트의 각 인덱스가 `list1[index] + list2[index]`에 해당하는 새 리스트를 반환하는 함수 `f23(list1, list2)`을 작성하세요.

`list1`과 `list2`의 길이가 같다고 가정합니다.

```
>>> f23([],[])  
[]  
>>> f23([1,2,3],[1,2,3])  
[2,4,6]  
>>> f23([0,0,0],[1,2,3])  
[1,2,3]
```

```
def f23(list1, list2):  
    list3 = []  
    for i in range( 0, len(list1) ) :  
        list3.append( list1[i] + list2[i] )  
    return list3
```


24.2 또는 3의 배수인 1에서 n까지의 모든 숫자를 출력하는 함수 f24(n)를 작성하세요.

```
>>> f24(10)
```

```
2
```

```
3
```

```
4
```

```
6
```

```
8
```

```
9
```

```
10
```

```
>>> f24(1)
```

```
>>> f24(3)
```

```
2
```

```
3
```

```
def f24(n):  
    for i in range( 1, n+1 ):  
        if i%2 == 0 or i%3 == 0 :  
            print (i)
```

25. 리스트에서 가장 큰 값을 반환하는 함수 f25(list)을 작성하세요(내포된 모든 리스트).
리스트는 중첩된 리스트입니다. 리스트가 비어 있지 않은 것으로 시작한다고 가정합니다.

```
>>> f25([[1,2,3],[4,5,6],[7,8,9]])
9
>>> f25([[3,2,1],[0,-1,-2]])
3
>>> f25([[1,2,3,4],[],[34],[],[56],[67]])
67
```

```
def f25(list):
    largest = list[0][0]
    for i in range( 0, len(list) ) :
        for j in range( 0, len(list[i])):
            if list[i][j] > largest :
                largest = list[i][j]
    return largest
```

26. 리스트에서 두 번째로 큰 값을 반환하는 함수 f26(list)를 작성하세요.

리스트의 요소가 모두 고유하고 두 개 이상의 요소가 포함되어 있다고 가정합니다.

```
>>> f26([1,4,3,2,5])
```

```
4
```

```
>>> f26([3,2])
```

```
2
```

```
>>> f26([3,4])
```

```
3
```

```
def f26(list):  
    largest = max( list[0],list[1] )  
    second_largest = min( list[0], list[1] )  
    for i in range( 2, len(list) ) :  
        if list[i] > largest :  
            second_largest = largest  
            largest = list[i]  
        elif list[i] > second_largest :  
            second_largest = list[i]  
    return second_largest
```

앞 슬라이드의 코드대로 하면, 결과값이 정상 출력되지 않습니다.
따라서 위의 'range(0,len(list))' 부분을 range(2,len(list))로 수정하였습니다.

27. 가장 왼쪽 자릿수를 반환하는 함수 f27(n)를 n으로 작성하세요. n이 양수라고 가정합니다.

```
>>> f27(1234)
```

```
1
```

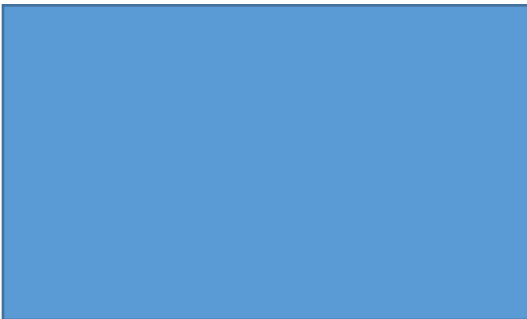
```
>>> f27(4321)
```

```
4
```

```
>>> f27(3)
```

```
3
```

```
def f27(n):  
    q = n  
    while q >= 10 :  
        q = q // 10  
    return q
```



28. 주어진 리스트에서 각 중첩된 리스트의 가장 큰 값을 인쇄할 함수 f28(list)을 작성하세요.
리스트는 중첩된 리스트입니다. 지정된 리스트의 각 중첩 리스트가 비어 있지 않다고 가정합니다.

```
>>> f28([[1,2,3],[4,5,6],[7,8,9]])
3
6
9
>>> f28([[3,2,1],[0,-1,-2]])
3
0
>>> f28([[1,2,3,4],[1],[34],[2],[3],[56],[67]])
4
1
34
2
3
56
67
```

```
def f28(list):
    for i in range( 0, len(list)):
        max_value = list[i][0]
        for j in range( 0, len(list[i]) ) :
            if max_value < list[i][j] :
                max_value = list[i][j]
        print ( max_value )
```

