11장. 컬렉션 관리

파이썬 정복

Contents

❖ 목차

1. 컬렉션 관리 함수

❖ enumerate

- 순서값과 요소값을 한꺼번에 구하는 내장함수

- Ex) 여러 학생의 성적을 출력할 경우
  - for문 이용
    - 순서값을 알 수 없음
    - 별도 변수 활용한 복잡한 과정 통해 순서값 구함

sequence

```
score = [ 88, 95, 70, 100, 99 ]
for s in score:
    print("성적 :", s)
```

sequence3

```
score = [ 88, 95, 70, 100, 99 ]
for no in range(len(score)):
    print(str(no + 1) + "번 학생의 성적 :", score[no])
```

1. 컬렉션 관리 함수

❖ zip

- 여러 개 컬렉션 합쳐 하나로 만듦
- 두 리스트의 대응되는 요소끼리 짝지어 튜플 리스트 생성

zip

```
yoil = ["월", "화", "수", "목","금", "토", "일"]
food = ["갈비탕", "순대국", "칼국수", "삼겹살"]
menu = zip(yoil, food)
for y, f in menu:
    print("%s요일 메뉴 : %s" % (y, f))
```

실행결과

```
월요일 메뉴 : 갈비탕
화요일 메뉴 : 순대국
수요일 메뉴 : 칼국수
목요일 메뉴 : 삼겹살
```

- 합쳐지는 두 리스트의 길이는 무관
- 생성되는 튜플의 순서는 원본 리스트의 순서와 같음

2. 람다 함수

❖ filter 함수

- 리스트 요소 중 조건에 맞는 것만을 골라냄
- 첫 번째 인수 : 조건 지정하는 함수
- 두 번째 인수 : 대상 리스트

filter

```
def flunk(s):
    return s < 60

score = [ 45, 89, 72, 53, 94 ]
for s in filter(flunk, score):
    print(s)
```

실행결과

```
45
53
```

- flunk 함수
  - 점수 s를 인수로 받아 60 미만인지 조사

2. 람다 함수

❖ map 함수

- 모든 요소에 대한 변환함수 호출, 새 요소값으로 구성된 리스트 생성
- 인수 구조는 filter 함수와 동일

map

```
def half(s):
    return s / 2

score = [ 45, 89, 72, 53, 94 ]
for s in map(half, score):
    print(s, end = ', ')
```

실행결과

```
22.5, 44.5, 36.0, 26.5, 47.0,
```

- half 함수
  - 인수로 전달받은 s를 절반으로 나누어 리턴

2. 람다 함수

❖ 람다 함수

- 이름 없고 입력과 출력만으로 함수 정의하는 축약된 방법
- lamda 인수:식
  - 인수는 여러 개 가질 수 있음

```
lambda x:x + 1
```

lambda

```
score = [ 45, 89, 72, 53, 94 ]
for s in filter(lambda x:x < 60, score):
    print(s)
```

3. 컬렉션의 사본

❖ 리스트의 사본

- 기본형 변수는 대입 이후 둘 중 하나 바꾸어도 다른 쪽에 영향 없음
- 컬렉션의 경우, 같은 리스트를 두 변수가 가리키는 것이라 영향 있음

varcopy

```
a = 3
b = a
print("a = %d, b = %d" % (a, b))

a = 5
print("a = %d, b = %d" % (a, b))
```

실행결과

```
a = 3, b = 3
a = 5, b = 3
```

listcopy

```
list1 = [ 1, 2, 3 ]
list2 = list1

list2[1] = 100
print(list1)
print(list2)
```

실행결과

```
[1, 100, 3]
[1, 100, 3]
```

3. 컬렉션의 사본

- copy 메서드로 두 리스트를 완전히 독립 사본으로 만들 수 있음

listcopy2

```
list1 = [ 1, 2, 3 ]
list2 = list1.copy()

list2[1] = 100
print(list1)
print(list2)
```

실행결과

```
[1, 2, 3]
[1, 100, 3]
```

- list[:] 으로 전체 범위에 대한 사본 만드는 방법도 가능

3. 컬렉션의 사본

deepcopy

```
list0 = [ 'a', 'b' ]
list1 = [ list0, 1, 2 ]
list2 = list1.copy()

list2[0][1] = 'c'
print(list1)
print(list2)
```

실행결과

```
[['a', 'c'], 1, 2]
[['a', 'c'], 1, 2]
```

deepcopy2

```
import copy

list0 = [ "a", "b" ]
list1 = [ list0, 1, 2 ]
list2 = copy.deepcopy(list1)

list2[0][1] = "c"
print(list1)
print(list2)
```

실행결과

```
[['a', 'b'], 1, 2]
[['a', 'c'], 1, 2]
```

3. 컬렉션의 사본

❖ is 연산자

- is 구문 통해 두 변수가 같은 객체 가리키는지 조사

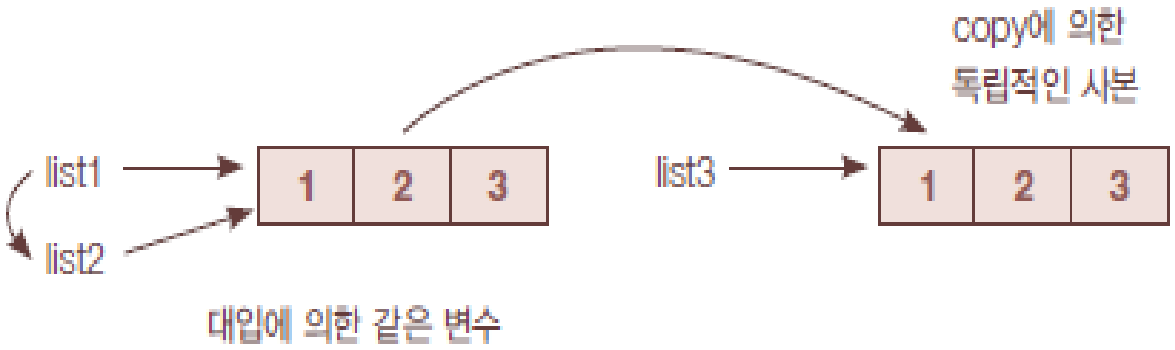
is

```
list1 = [ 1, 2, 3 ]
list2 = list1
list3 = list1.copy()

print("1 == 2" , list1 is list2)
print("1 == 3" , list1 is list3)
print("2 == 3" , list2 is list3)
```

실행결과

```
1 == 2 True
1 == 3 False
2 == 3 False
```



Thank You !

파이썬 정복