

# 14. 컬렉션 관리

## 컬렉션 함수

### - enumerate()

. 데이터 들을 순환 하며 읽어내도록 하며 순서 값과 요소 값 을 한꺼번에 구해주는 내장함수

```
7 score = [88,95,70,100,99]
8 for no,value in enumerate(score,1) :
9     print(no, ' 번의 학생 성적 : ', value)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Python\Source> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe C:\Python\Source\11.collection[컬렉션관리].py"
1 번의 학생 성적 : 88
2 번의 학생 성적 : 95
3 번의 학생 성적 : 70
4 번의 학생 성적 : 100
5 번의 학생 성적 : 99
```

> enumerate() 내장 함수 를 사용하여 리스트 의 데이터와 함께 추출한 순번을 반환 받을 수 있다 , (추출할 내용, 시작 숫자)

> no : 순서 , value : 리스트에서 추출한 값

### - zip()

. 여러 개 의 컬렉션을 합쳐 하나로 만든다.

> 합쳐진 결과는 순차적으로 반환 된다.

> zip() 형태의 자료형 은 내용을 바로 출력하여 확인 할 수 없다.

```
Chap01_intro > 11.collection[컬렉션관리].py > ...
1 yoil = ['월','화','수','목','금','토','일']
2 food = ['갈비탕','순대국','칼국수','삼겹살']
3 menu = zip(yoil,food)
4 for y,f in menu:
5     print('%s 요일 메뉴 : %s' % (y,f))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Source/11.collection[컬렉션관리].py
월 요일 메뉴 : 갈비탕
화 요일 메뉴 : 순대국
수 요일 메뉴 : 칼국수
목 요일 메뉴 : 삼겹살
```

\* 쌍이 맞는 개수 만큼만 반환

. Dict() 를 이용하면 사전을 생성 한다.

```
Chap01_intro > 11.collection[컬렉션관리].py > ...
1 yoil = ['월','화','수','목','금','토','일']
2 food = ['갈비탕','순대국','칼국수','삼겹살']
3 menu = dict(zip(yoil,food))
4 print(menu)
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Source/11.collection[컬렉션관리].py
{'월': '갈비탕', '화': '순대국', '수': '칼국수', '목': '삼겹살'}
```

> yoil 은 Key 로 , food 는 value 로 매핑 된다.

. Any() 와 All()

- > any() : 하나라도 True 이면 True
- > all() : 하나라도 다른값이 있으면 false

```
Chap01_intro > 11.collection[컬렉션관리].py > ...
1  adult = [True,False,True,False]
2  print(any(adult))
3  print(all(adult))
4
5

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/py
True
False
```

### 메서드를 인수로 받는 함수

- filter ( 함수 , 리스트 )

. 리스트 의 요소 중 조건에 맞는것만 추출하여 리스트 를 작성한후 반환한다 ( 함수를 한 True / False 의 결과 를 list 로 생성한다 )

- > 첫번째 인수 는 함수를 인수 로 전달 (대리자)
- > 두번째 인수 는 리스트 를 인수 로 전달

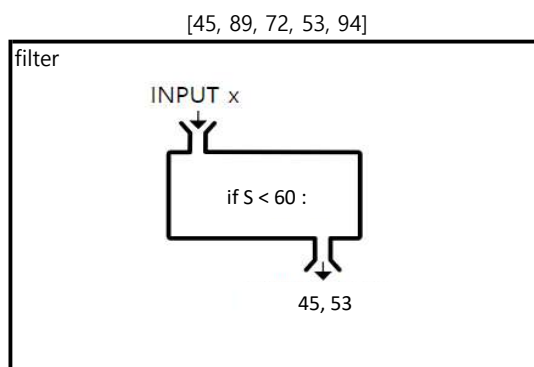
```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1  def flunk(s):
2      return s < 60
3
4  score = [45,89,72,53,94]
5  for s in filter (flunk, score):
6      print(s)
7

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/py
45
53
```

- >5행 : filter 함수 에 flunk 함수 와 score 리스트 를 인수 로 전달 한다.
- >6행 : score 리스트 에서 하나씩 추출 하여 flunk 의 인수로 전달 하고 그 결과 가 참 인 것만 출력 한다.



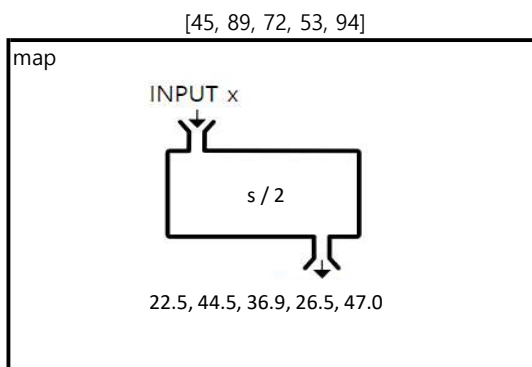
- map()
  - . 모든 요소에 대해 변환 함수를 호출 하여 새 요소값으로 구성 된 리스트를 생성
  - . Filter 함수 와 동일한 인수 구조 를 가진다. (함수를 통한 연산 결과 를 list 로 생성한다)

```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1  def half(s):
2      return s / 2
3
4  score = [45,89,72,53,94]
5  for s in map(half, score):
6      print(s,end=', ')
7
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/pyt
22.5, 44.5, 36.0, 26.5, 47.0,
```

> 리스트 의 각 원소를 하나씩 추출 하여 half 메서드 의 결과 를 출력한다.



- . 두개 이상의 리스트 를 받아 각 리스트 의 요소 를 조합
  - > 두개 의 요소값을 전달 받으므로 리스트 는 두개 여야 한다.

```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1  def half(s,b):
2      return s+b
3
4  score = [45,89,72,53,94]
5  bonus = [2, 3, 0, 5, 10]
6  for result in map(half, score,bonus):
7      print(result,end=', ')
8
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/pyth
47, 92, 72, 58, 104,
```

> 리스트 의 원소 가 쌍을 이루는 요소 만 적용된다.

```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1  def half(s,b):
2      return s+b
3
4  score = [45,89,72,53,94]
5  bonus = [2, 3, 0]
6  for result in map(half, score,bonus):
7      print(result,end=', ')
8
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/py
47, 92, 72,
```

## 람다 함수

- 이름이 없는 메서드 를 간편하게 작성하여 호출하는 문법
  - . 재사용을 하지 않는 로직일 경우 메서드를 굳이 생성 할 필요 가 없다.
  - . 이때 전달하는 로직을 이름이 없는 메서드 라고 하여 무명 메서드 라 부른다
  - . 무명 메서드의 로직을 간편히 구성하여 전달 하는데 이를 람다 라고 한다.
- . 함수, 메서드 의 시그니처를 간략 하게 표현하여 1회성의 함수,메서드 를 만들어 호출할때 사용한다.
  - > 시그니처 : 함수 또는 메서드 를 구성하는 모양, 틀
  - > def flunk(s):  
    return s < 60 : 인수를 1개 받아 bool 데이터유형을 반환하는 함수
  - > lambda X : X < 60

```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1 score = [45,89,72,53,94]
2 for s in filter(lambda x:x < 60 , score) :
3     print(s)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Py  
45  
53

- > lambda x : x < 60
  - \* X 라는 인자를 받아 X < 60 의 결과 (True / False) 를 반환하는 무명 메서드 로직(lambda) 을 전달함

- . 위 의 map 함수 를 lambda 로 표현한 결과

```
Chap01_intro > 12.CollectionFunction[람다함수].py > ...
1 score = [45,89,72,53,94]
2 for s in map(lambda x:x / 2 , score) :
3     print(s)
4
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/pyt  
22.5  
44.5  
36.0  
26.5  
47.0

## 실습

1. 위의 예제 중 리스트 를 두개 받아 각 요소를 합하여 결과를 반환하는 map 예제를 lambda 를 이용하여 표현 해 보세요.

```
47  
92  
72  
58  
104
```

## 컬렉션의 사본

### - 리스트의 얇은 복사

. 리스트 직접 할당 할 경우 참조 형식으로 데이터의 주소를 전달 하므로 값은 공유를 한다.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1 score = [45,89,72,53,94]
2 score_bak = score
3 score_bak[0] = 90
4 print(score_bak)
5 print(score)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python  
[90, 89, 72, 53, 94]  
[90, 89, 72, 53, 94]

> 복사를 한 score\_bak 리스트의 0 번째 값을 90 으로 변경 하였는데 score 의 데이터도 같이 변경 된것을 확인 할 수 있다.

> 1. Stack 에 score 리스트 객체 생성 및 heap 에 리스트 데이터 등록

. Heap 에 등록된 데이터 는 각 데이터 별로 메모리 주소를 가진다. (100,200,300,400,500)

. Score 객체 는 Heap 의 각 메모리 주소를 참조 한다.

stack
score [100,200,300,400,500]

Heap			
100	45	400	89
200	72	500	53
300	94		

> 2. Score\_bak 리스트 객체 생성 및 score 가 참조하고 있는 주소 전달

. Score\_bak 객체도 heap 의 같은 메모리 주소 에 있는 데이터 를 참조 한다.

stack
score_bak [100,200,300,400,500]
score [100,200,300,400,500]

Heap			
100	45	400	89
200	72	500	53
300	94		

> 3. Score\_bak 객체가 바라보고있는 0번째 메모리 의 값을 90으로 변경

stack
score_bak [100,200,300,400,500]
score [100,200,300,400,500]

Heap			
100	90	400	89
200	72	500	53
300	94		

> 4. score 출력

. score 객체는 heap 메모리 번지의 값을 그대로 호출 하므로 score\_bak 객체로 변경 한 값을 그대로 출력한다.

```
[90, 89, 72, 53, 94]
```

### - 리스트의 깊은 복사 1 (copy())

. 원본 데이터 와는 별개로 다른 메모리 번지에 값을 그대로 복사하여 새로운 메모리 번지를 참조한다.

> copy() 메서드 를 이용하여 복사본을 생성 한다.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1 score = [45,89,72,53,94]
2 score_bak = score.copy()
3 score_bak[0] = 90
4 print(score_bak)
5 print(score)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python  
[90, 89, 72, 53, 94]  
[45, 89, 72, 53, 94]

\* 원본 값에는 영향을 주지 않는 새로운 메모리 값을 참조하고있는 복사본이 만들어 진다.

- 리스트의 깊은 복사 2 ( [: ])
  - . [: ] 도 copy() 메서드와 같은 기능을 한다.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1  score = [45,89,72,53,94]
2  score_bak = score[:]
3  score_bak[0] = 90
4  print(score_bak)
5  print(score)
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
[90, 89, 72, 53, 94]
[45, 89, 72, 53, 94]
```

- 중첩 리스트의 깊은 복사 (deepcopy)
  - . 중첩 리스트의 경우 copy (또는 [: ])로는 완전히 복사 할 수 없다.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1  score = [[10,20,30],89,72,53,94]
2  score_bak = score.copy()
3  score_bak[0][1] = 90
4  print(score_bak)
5  print(score)
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
[[10, 90, 30], 89, 72, 53, 94]
[[10, 90, 30], 89, 72, 53, 94]
```

- >1행 : 0 번 index 에 리스트 가 등록된 2중 리스트 score
- >2행 : copy() 메서드 로 깊은 복사 시도
- >3행 : score\_bak 리스트 의 [0],[1] 인덱스 위치의 데이터를 90으로 변경
- >5행 : 원본 리스트 까지 변경된 것을 확인 할 수 있다.

- > Copy 외부 모듈을 import ( 외부에서 만들어 둔 기능을 사용 할 수 있도록 배포되는 Copy 라이브러리 를 참조 한다)
  - . Copy.deepcopy() 메서드 를 이용하여 중첩 리스트 의 내용도 깊은 복사 처리

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1  import copy
2
3  score = [[10,20,30],89,72,53,94]
4  score_bak = copy.deepcopy(score)
5  score_bak[0][1] = 90
6  print(score_bak)
7  print(score)
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
[[10, 90, 30], 89, 72, 53, 94]
[[10, 20, 30], 89, 72, 53, 94]
```



- is 연산자

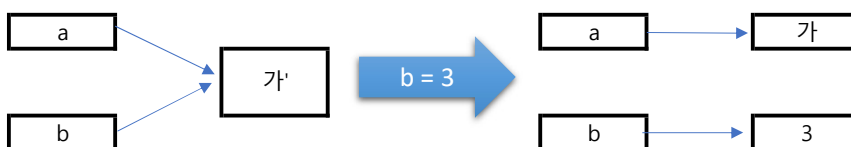
. 두 객체 가 같은 메모리를 참조 하고 있는 지를 확인하는 연산자.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1 list1 = [1,2,3]
2 list2 = list1
3 list3 = list1.copy()
4
5 print('list1 과 list2 이 같은 메모리를 참조하는가? %s' % (list2 is list1))
6 print('list2 와 list3 이 같은 메모리를 참조하는가? %s' % (list2 is list3))
7 print('list1 과 list3 이 같은 메모리를 참조하는가? %s' % (list1 is list3))
-
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.
D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe "d:/Python/Chap01_intro/13.CollectionFunction[
list1 과 list2 이 같은 메모리를 참조하는가? True
list2 와 list3 이 같은 메모리를 참조하는가? False
list1 과 list3 이 같은 메모리를 참조하는가? False
```

- 정수형 변수 의 복사 \*\*\*\*\*

. 변수 는 일시적으로 같은 메모리 를 참조 하지만 다른 값을 할당 할 경우 메모리 번지를 다른곳을 참조하기 시작한다.

```
Chap01_intro > 13.CollectionFunction[컬렉션의 사본].py > ...
1 a = '가'
2 b = a
3 print('a 와 b 는 같은 메모리를 참조하는가? %s' % (a is b))
4 b = 3
5 print('a 와 b 는 같은 메모리를 참조하는가? %s' % (a is b))
6
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.
D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe "d:/Python/Chap01_int
a 와 b 는 같은 메모리를 참조하는가? True
a 와 b 는 같은 메모리를 참조하는가? False
```



## 실습

2. 1 ~ 10 정수의 리스트 a 와 각 정수의 제곱 리스트 b 를 만든 후 두 리스트 를 합쳐 사전으로 만든후 사전에서 6의 제곱값을 검색하여 출력하는 로직을 구현해 보세요

```
dict_items([(1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49), (8, 64), (9, 81), (10, 100)])
36
```

3. price 리스트에 상품 가격 네개가 저장 되어 있다. 함수 와 리스트를 map 을 사용하여 모든 상품 가격을 20% 세일한 값으로 출력 하는 로직을 구현해 보세요

```
800.0, 640.0, 1600.0, 2800.0,
```