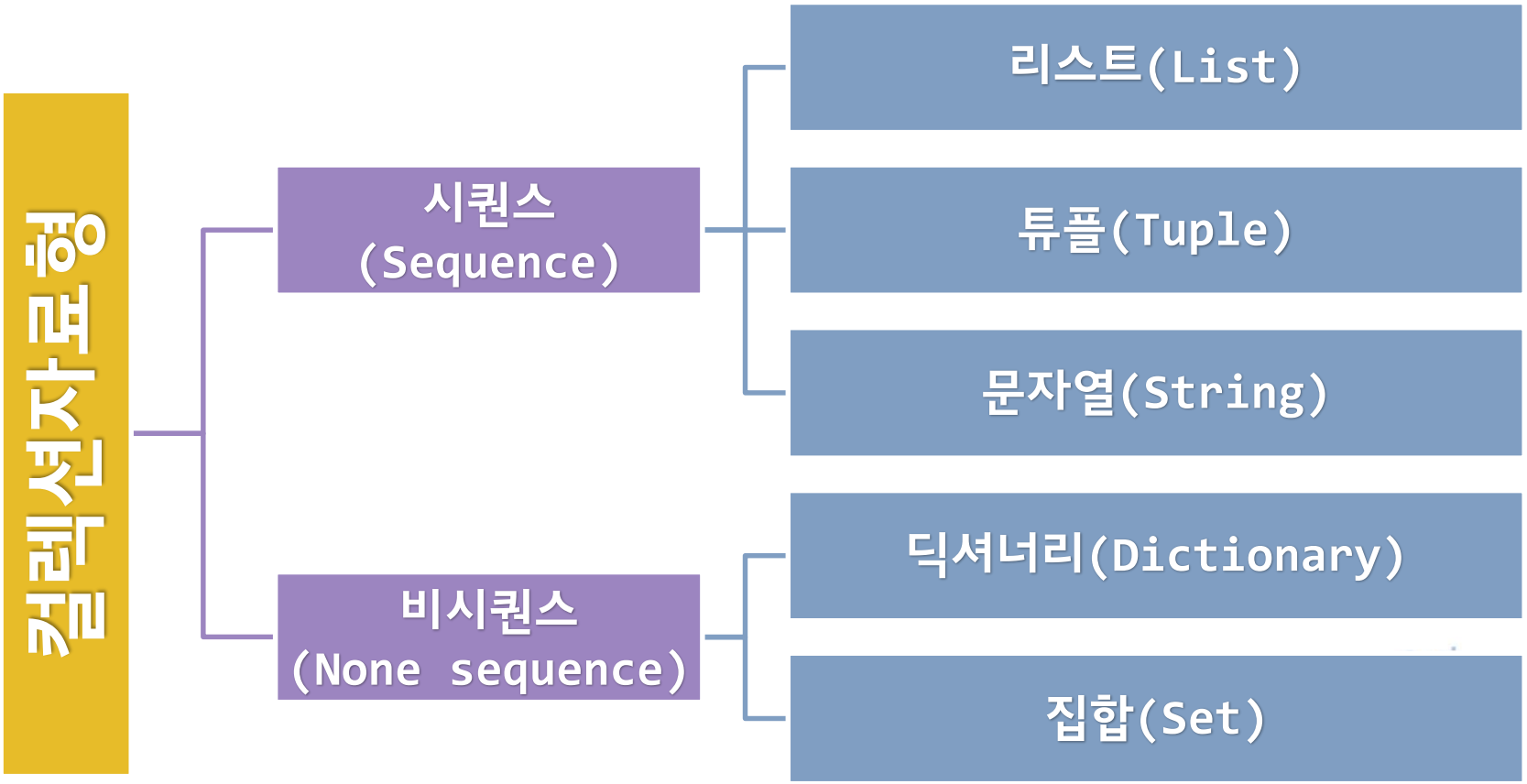
Python
04. 컬렉션 데이터타입

컬렉션자료형(collection data type)

- 여러 요소를 묶어서 처리할 수 있는 자료형
- 각 요소에 접근하는 방법이 제공



리스트(list)

- 특징
  - 리스트의 각 요소는 순서를 가지고 있음
  - 대괄호([])로 작성되며 각 요소는 쉼표(,)로 구분
  - 요소를 추가, 수정, 삭제 가능
- 리스트 생성
  - []나 리스트 함수
- 인덱싱
  - 요소의 선택은 0부터 시작되는 인덱스로 접근
  - 리스트명[인덱스]
- 슬라이싱
  - 요소의 일부분 선택
  - 리스트명[시작인덱스:종료인덱스] :
    시작인덱스부터 종료인덱스-1 까지
- 리스트 연산
  - + 연산 : 리스트와 리스트 연결
  - * 연산 : 리스트 요소 반복

```
# 리스트 생성 : []나 리스트 함수
lst = []
lst = list()
lst = [10, 30, 20]

# 인덱싱 : 요소의 선택은 0부터 시작되는 인덱스로 접근
print(f'첫번째 요소 :{lst[0]}')
print(f'마지막 요소 :{lst[-1]}')

# 슬라이싱 : 요소의 일부분 선택
print(lst[:2])

# 리스트 연산
# + 연산 : 리스트와 리스트 연결
# * 연산 : 리스트 요소 반복
lst2 = [100, 200]
lst = lst + lst2
print(lst)

lst2 = lst2 * 3
print(lst2)
```

```
[]
[]
[10, 30, 20]
첫번째 요소 :10
마지막 요소 :20
[10, 30]
[10, 30, 20, 100, 200]
[100, 200, 100, 200, 100, 200]
```

리스트(list)

- **리스트 추가**
  - 리스트명.append(추가요소) : 마지막에 추가
  - 리스트명.insert(인덱스, 추가요소) : 해당 인덱스에 요소 추가
- **리스트 삭제**
  - 리스트명.pop() : 마지막 요소 삭제
  - 리스트명.pop(인덱스) : 해당 인덱스 요소 삭제
  - 리스트명.remove(요소값) : 해당 요소값 삭제
    - 해당 값이 없으면 오류
- **리스트 수정**
  - 리스트명[인덱스]=변경값

```
# 리스트 추가
# 리스트명.append(추가요소) : 마지막에 추가
# 리스트명.insert(인덱스, 추가요소) : 해당 인덱스에 요소 추가
lst.append('a')
lst.insert(0,'b')

print("리스트 추가:" , lst)

# 리스트 삭제
# 리스트명.pop() : 마지막 요소 삭제
# 리스트명.pop(인덱스) : 해당 인덱스 요소 삭제
# 리스트명.remove(요소값) : 해당 요소값 삭제 해당 값이 없으면 오류
lst.pop()
lst.pop(0)
lst.remove(100)
lst.remove(200)
print(f"리스트 삭제:{lst}")

# 리스트 수정
# 리스트명[인덱스]=변경값
lst[0] = 100
print(f"리스트 수정:{lst}")
```

```
리스트 추가: ['b', 10, 30, 20, 100, 200, 'a']
리스트 삭제:[10, 30, 20]
리스트 수정:[100, 30, 20]
```

리스트(list)

- **리스트 위치 반환**
  - 리스트명.index(요소값)
  - 해당 요소값의 위치를 반환
  - 여러 개 요소가 존재하면 첫번째 위치 반환하고 없으면 오류
- **리스트 요소 개수 구하기**
  - 리스트명.count(요소값)
- **리스트 정렬**
  - 리스트명.sort() : 오름차순
  - 리스트명.sort(reverse=True): 내림차순
- **리스트 뒤집기**
  - 리스트명.revers() : 요소를 역순으로 뒤집어 줌

```
# 리스트 위치 반환
# 리스트명.index(요소값) : 해당 요소값의 위치를 반환, 없으면 오류
print(f'20의 위치 : {lst.index(20)}')

# 리스트 요소 개수 구하기
# 리스트명.count(요소값)
print(f'20의 개수 : {lst.count(20)}')

# 리스트 정렬
# 리스트명.sort() : 오름차순
# 리스트명.sort(reverse=True): 내림차순
lst.sort()
print(f"리스트 정렬:{lst}")

# 리스트 뒤집기
# 리스트명.revers() : 요소를 역순으로 뒤집어 줌
lst.reverse()
print(f"리스트 뒤집기:{lst}")
```

```
20의 위치 : 2
20의 개수 : 1
리스트 정렬:[20, 30, 100]
리스트 뒤집기:[100, 30, 20]
```

튜플(tuple)

- 특징
  - 튜플의 각 요소는 순서를 가지고 있음
  - 괄호(())로 작성되며 각 요소는 쉼표(,)로 구분
  - 요소를 추가, 수정, 삭제 불가능
    - 한번 결정된 요소는 변경 불가능
  - 인덱싱과 슬라이싱은 리스트와 동일
  - +와 *연산은 리스트와 동일
  - 각 요소에 변수 할당 가능
    - x, y = (10, 20)

```
#튜플
tp = (10, 20, 30)

# 인덱싱과 슬라이싱은 리스트와 동일
print(tp[0])
print(tp[:2])

# +와 *연산은 리스트와 동일
tp = tp*3
print(tp)

# 각 요소에 변수 할당 가능
tpx, tpy = (10, 20)
print(f'typx = {tpx}, tpy = {tpy}')

# 요소를 추가, 수정, 삭제 불가능
# 한번 결정된 요소는 변경 불가능
tp[0] = 100
```

```
10
(10, 20)
(10, 20, 30, 10, 20, 30, 10, 20, 30)
typx = 10, tpy = 20
Traceback (most recent call last):
  File "
    tp[0] = 100
TypeError: 'tuple' object does not support item assignment
```

딕션너리(dictionary)

- 특징
  - 키(key)와 값(value) 쌍을 요소로 가짐
  - 중괄호({})로 작성되며 각 요소는 쉼표(,)로 구분
  - 순서 없음
  - 키는 변경할 수 없으며 값은 변경가능
- 딕션너리 접근
  - 특정 요소 값 : 딕션너리명[키]
  - 요소 키 컬렉션 : 딕션너리명.keys()
  - 요소 값 컬렉션 : 딕션너리명.values()
  - 키와 요소 컬렉션은 list() 생성자로 리스트로 변환가능
- 요소 추가
  - 딕션너리명[추가키]=값
- 요소 수정
  - 딕션너리명[키]=값
- 요소 삭제
  - del 딕션너리명[키]
  - 딕션너리명.pop[키]

```
#딕션너리
dt = {'a':200, 'b':150, 'c':100}

#딕션너리 키
print(f'딕션너리 키 : {list(dt.keys())}')

#딕션너리 값
print(f'딕션너리 값 : {list(dt.values())}')

#딕션너리 접근
print(f'딕션너리 a 값 : {dt["a"]}')

#딕션너리 추가
dt['d'] = 30
print(dt)

#딕션너리 수정
dt['b'] = 1
print(dt)

#딕션너리 삭제
del dt['b']
print(dt)
```

```
딕션너리 키 : ['a', 'b', 'c']
딕션너리 값 : [200, 150, 100]
딕션너리 a 값 : 200
{'a': 200, 'b': 150, 'c': 100, 'd': 30}
{'a': 200, 'b': 1, 'c': 100, 'd': 30}
{'a': 200, 'c': 100, 'd': 30}
```

집합(set)

- 특징
  - 중복 없는 요소로만 구성
  - 중괄호({}) 작성되며 각 요소는 쉼표(,)로 구분
  - 순서 없음
- 생성자 set()
  - 리스트나 튜플의 중복 요소 제거
- 요소 추가
  - 집합명.add(요소)
    - 하나의 요소 추가
  - 집합명.update(추가집합)
    - 여러 요소를 추가
- 요소 삭제
  - 집합명.remove(요소값)
    - 하나의 요소 삭제
  - 집합명.clear()
    - 모든 요소 삭제
- 집합 연산
  - 교집합 : &
  - 합집합 : |
  - 차집합 : -

```
#집합
a = (1, 2, 2, 3)
b = [10, 20, 30, 30]

#중복요소제거 및 형변환
print(f'{a}의 type : {type(a)}')
a = set(a)
print(f'{a}의 type : {type(a)}')

print(f'{b}의 type : {type(b)}')
b = set(b)
print(f'{b}의 type : {type(b)}')

#요소 추가
a.add(100)
b.update({100,200,300})

print(a)
print(b)

#요소 삭제
a.remove(100)
print(a)

a.clear()
print(a)

#집합 연산
a.update({100,200,500})
print(f'a집합 = {a} , b집합 = {b}')
print(f'합집합 = {a&b}')
print(f'교집합 = {a|b}')
print(f'차집합 = {a-b}')
```

```
(1, 2, 2, 3)의 type : <class 'tuple'>
{1, 2, 3}의 type : <class 'set'>
[10, 20, 30, 30]의 type : <class 'list'>
{10, 20, 30}의 type : <class 'set'>
{1, 2, 3, 100}
{20, 100, 200, 10, 300, 30}
{1, 2, 3}
set()
a집합 = {200, 100, 500} , b집합 = {20, 100, 200, 10, 300, 30}
합집합 = {200, 100}
교집합 = {100, 200, 10, 300, 500, 20, 30}
차집합 = {500}
```

컬렉션 데이터

- 컬렉션 데이터 모든 요소에 접근
  - for 요소변수 in 컬렉션명 :
- 내장함수
  - len(컬렉션명)
    - 컬렉션 요소 개수
  - max(컬렉션명)
    - 가장 큰 요소 값 반환
    - 딕션너리인 경우는 키 값 중 가장 큰 요소 반환
  - min(컬렉션명)
    - 가장 작은 요소 값 반환
    - 딕션너리인 경우는 키 값 중 가장 작은 요소 반환
  - sum(컬렉션명)
    - 수치 요소로 이루어진 컬렉션 요소의 합
    - 딕션너리인 경우는 키 값의 합
  - sorted(컬렉션명)
    - 컬렉션요소 정렬
    - 딕션너리인 경우는 키 값을 정렬하여 키 리스트 반환

```python
#컬렉션 요소 접근
lst = [10,50,30]
tp = ('a','b','c')

for item in lst :
    print(item, end=' ')
print()

for item in tp :
    print(item, end=' ')
print()

#딕션너리 만들기
dt = dict(zip(tp,lst))
print(dt)

for item in dt :
    print(f'{item} => {dt[item]}')

#내장함수
print(f'{lst}의 요소 개수 : {len(lst)}')
print(f'{lst}의 최대값 : {max(lst)}')
print(f'{lst}의 최소값 : {min(lst)}')
print(f'{lst}의 합계 : {sum(lst)}')
print(f'{lst}의 정렬 : {sorted(lst)}')
print(lst)
```

```
10 50 30
a b c
{'a': 10, 'b': 50, 'c': 30}
a => 10
b => 50
c => 30
[10, 50, 30]의 요소 개수 : 3
[10, 50, 30]의 최대값 : 50
[10, 50, 30]의 최소값 : 10
[10, 50, 30]의 합계 : 90
[10, 50, 30]의 정렬 : [10, 30, 50]
[10, 50, 30]
```

해결문제1

- 기상청에서 제공하는 2021년 3월 평균기온 정보를 참고하여, 다음을 구하시오.

- https://data.kma.go.kr/stcs/grnd/grnd_TaList.do?pgmNo=70

  - 최고 기온인 **날**
  - 최저 기온인 **날**
  - 3월 평균 **기온**

```
최고 기온 : 14.5 => 26일
최저 기온 : 5.1 => 5일
평균 기온 : 10.44도
```

해결문제2

- 기상청 공공데이터 사이트에서 2020년 1월에서 12월까지 부산 평균 기온을 조사하여 2020년의 전체 평균기온을 구하시오.

- https://data.kma.go.kr/stcs/grnd/grnd_TaList.do?pgmNo=70

해결문제3

- **2011~2020년 기온데이터(지역:부산)를 조사하여, 아래 내용을 출력하시오.**
- https://data.kma.go.kr/stcs/grnd/grnd_TaList.do?pgmNo=70

1. 홀수년도 출력
2. 년도와 평균기온 딕셔너리 만들기
3. 평균, 최고, 최저 기온 년도 – 온도 출력

평균기온 최소 2012년 – 14.5℃
최고기온 최대 2016년 – 19.6℃
최저기온 최소 2012년 – 18.6℃

년	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	
지점	159	159	159	159	159	159	159	159	159	159	
평균기온(℃)		14.6	14.5	15.3	15.1	15.4	15.7	15.2	15.1	15.7	15.2
최저기온(℃)		-12.8	-9.9	-10.7	-6	-7.8	-10.2	-7.7	-9.9	-4.4	-8
최고기온(℃)		33	34.5	35	32.9	33.5	37.3	36.2	36.4	35	33.2