순회형 자료형

- 기본 자료형
  - 문자열
- 복합 자료형
  - 리스트
  - 튜플
  - 딕셔너리
  - 세트

순회형 연산자

결합 연산자 **+**

- ***s* + *t***: 시퀀스형 ***s***와 ***t***를 합친 시퀀스형 자료를 반환한다.

In [ ]:

```
# 리스트와 리스트 더한 결과를 반환한다.
[1, 2, 3] + ['a', 'b', 'c']
```

반복 결합 연산자 *

- ***s* * *n***: 시퀀스형 ***s***를 ***n*** 횟수만큼 반복한 시퀀스형 자료를 반환한다.

In [ ]:

```
# 튜플을 5번 반복한 새로운 튜플을 반환한다.
(1, 2, 3) * 5
```

멤버십 연산자 **in/not in**

- ***x* in *i***: 객체 ***x***가 순회형 ***i***에 있으면 참(**True**)을 반환한다.
- **not in**은 객체 ***x***가 순회형 ***i***에 없으면 참(**True**)을 반환한다.

In [ ]:

```
# 집합 안에 'a'가 있는지 확인한다.
'a' in {'a', 'b', 'c'}
```

In [ ]:

```
# 집합 안에 'x'가 없는지 확인한다.
'x' not in {'a', 'b', 'c'}
```

순회형 함수

질의 함수 : **len()**, **all()**, **any()**
연산 함수 : **min()**, **max()**, **sum()**
정렬 함수 : **reversed()**, **sorted()**
순환문과 함께 사용하는 유용한 함수 : **range()**, **enumerater()**, **zip()**

질의 함수

len(*x*)

- ***x***의 길이를 반환한다.
- ***x***가 복합자료형이면 객체의 총 개수를 반환한다.
- ***x***가 문자열이면 문자의 개수를 반환한다.

In [ ]:

```
# 리스트의 길이(즉, 담고 있는 객체의 수)를 반환한다.
len([1, 2, 3])
```

In [ ]:

```
# 문자열의 길이(즉, 문자의 개수)를 반환한다.
len('Python')
```

all(*i*)

- 순회형 ***i***의 모든 객체가 '참'일 때만 참(**True**)을 반환한다.

In [ ]:

```
x = [7, -5, 8, 3, 9, 0]
all(x)
```

0이 거짓이라 **False**를 반환한다.

any(*i*)

- 순회형 ***i*** 중 한 객체라도 '참'이면 참(True)을 반환한다.

In [ ]:

```
# --- [7, -5, 8, 3, 9, 0]
any(x)
```

앞에서 불린을 설명할 때

- '거짓'의 값은 **False**, **0**, **0.0**, **None**, 빈 문자열(''), 빈 복합자료형(**[]**, **()**, **{}**, **set()**)이라고 했는데

이를 **any()**나 **all()** 함수로 간단히 확인해볼 수 있다.

In [ ]:

```
# 불린형에서 False로 처리되는 것들이다.
any([False, 0, 0.0, None, '', [], (), set(), {}])
```

In [ ]:

```
all([-15, 1.34, [1], (1,), set('a'), {None: 'NaN'}, ''])
```

빈 문자열이 있다.

연산 함수

min()과 **max()**는 순회형의 값 중 최솟값과 최댓값을 반환하는 함수다.

min(*i*, *key*)

- 순회형 ***i***의 객체 중 가장 작은 값을 가진 객체를 반환한다.
- ***key*** 전달인자가 주어질 경우 전달인자 함수로 처리한 결괏값 중 가장 작은 값을 가진 객체를 반환한다.

In [ ]:

```
x = [7, -5, 8, 3, 9]
# x 중 최댓값을 찾아서 반환한다.
max(x)
```

max(*i*, *key*)

- 순회형 ***i***의 객체 중 가장 큰 값을 가진 객체를 반환한다.
- ***key*** 전달인자가 주어질 경우 전달인자 함수로 처리한 결괏값 중 가장 큰 값을 가진 객체를 반환한다.

In [ ]:

```
# --- x = [7, -5, 8, 3, 9]
# x 중 최솟값을 찾아서 반환한다.
min(x)
```

min()과 **max()** 함수는 ***key***의 전달인자로 함수를 사용할 수 있다.

이때 함수 이름만 사용하며, 함수 뒤에 붙는 괄호는 생략한다.

다음 코드는 ***key***로 전달하는 함수로 **abs()**를 사용한다.

In [ ]:

```
# --- x = [7, -5, 8, 3, 9]
# x의 절대 값 중에 최솟값을 찾아서 반환한다.
min(x, key=abs)
```

min()과 **max()** 함수에 숫자가 아닌 문자도 올 수 있다.

In [ ]:

```
s = ['a', 'B', 'c', 'd', 'E']
# s 중 최댓값을 찾아서 반환한다.
max(s)
```

대소문자를 구분하지 않고 최댓값을 구하려면 다음처럼 문자열 클래스의 **lower()** 메소드를 불러 모두 소문자로 처리한 후 비교해서 최댓값 알파벳 문자를 찾는다.

In [ ]:

```
# --- s = ['a', 'B', 'c', 'd', 'E']
# s 중 대소문자 구분하지 않고 최댓값을 찾아서 반환한다.
max(s, key=str.lower)
```

sum(***i***, ***start***])

- 순회형 ***i***의 모든 객체를 더한 값을 반환한다.
- ***start*** 전달인자가 주어지지 않으면 기본값은 **0**이다.
- ***start*** 전달인자가 주어지면 ***start*** 전달인자와 순회형 ***i***의 객체들의 전체 합을 반환한다,
- ***i***에 문자열이 포함되어 있을 경우 **TypeError** 예외가 발생한다.

In [ ]:

```
x = [7, -5, 8, 3, 9]
# x의 모든 값을 더한 값을 반환한다.
sum(x)
```

In [ ]:

```
# 9와 x의 모든 값을 더한 값을 반환한다.
sum(x, 9)
```

정렬 함수

reversed(***seq***)

- 시퀀스형 ***seq***의 주어진 객체 순서를 역순으로 순회하는 순회자(iterator)를 반환한다.

In [ ]:

```
L = [7, -5, 8, 3, 9]
reversed(L)
```

In [ ]:

```
# reversed()가 반환한 객체의 자료형을 확인한다.
type(reversed(L))
```

In [ ]:

```
# reversed()가 반환한 객체를 리스트로 형변환한다.
list(reversed(L))
```

sorted(***i, key=None, reverse=False***)

- 순회형 ***i***의 객체를 정렬하는 방식의 순서로 바꾼 후 **리스트**로 반환한다.
- ***key*** 전달인자를 사용하면 DSU(Decorate, Sort, Undecorated) 정렬이 가능하다.
- ***reverse*** 전달인자가 참(**True**)이면 정렬이 역순으로 이루어진다.

sorted() 함수의 기본 정렬 방식은 오름차순이다.

In [ ]:

```
L = [7, -5, 8, 3, 9]
```

In [ ]:

```
# 오름차순으로 정렬한다.
sorted(L)
```

In [ ]:

```
# --- [7, -5, 8, 3, 9]
# 내림차순으로 정렬한다.
sorted(L, reverse=True)
```

In [ ]:

```
# --- [7, -5, 8, 3, 9]
# 절댓값으로 해서 오름차순으로 정렬한다.
sorted(L, key=abs)
```

In [ ]:

```
# --- [7, -5, 8, 3, 9]
# 절댓값으로 해서 내림차순으로 정렬한다.
sorted(L, key=abs, reverse=True)
```

순환문과 함께 사용하는 유용한 함수

range(***start, stop, step***)

- 정수 순회자를 반환한다.
- 한 개의 전달인자(***stop***)가 주어지면 순회자는 0부터 ***stop*** **- 1**까지의 정수를 반환한다.
- 두 개의 전달인자(***start, stop***)가 주어지면 순회자는 ***start***부터 ***stop*** **- 1**까지의 정수를 반환한다.
- 세 개의 전달인자(***start, stop, step***)가 주어지면 순회자는 ***start***부터 ***step***만큼의 간격을 두고 ***stop*** **- 1**까지의 정수를 반환한다.

In [ ]:

```
list1 = [x for x in range(10)]
list2 = [x for x in range(3, 10)]
list3 = [x for x in range(3, 10, 2)]

print(list1)
print(list2)
print(list3)
```

enumerate(***i, start=0***)

- 보통 **for ... in** 순환문과 함께 사용하며 **(인덱스, 객체)**의 튜플 쌍으로 순회할 수 있는 열거형(enumerate) 객체를 반환한다.
- ***start*** 전달인자가 주어지지 않으면 기본 값은 **0**이다.
- ***start*** 전달인자가 주어지면 ***start*** 전달인자부터 인덱스가 시작된다.

In [ ]:

```
fruits = ['사과', '바나나', '수박', '오렌지']

for i in fruits:
    print(i)
```

In [ ]:

```
for i in enumerate(fruits):
    print(i)
```

In [ ]:

```
for i, v in enumerate(fruits, 1):
    print(f'{i}번째 과일은 {v}입니다.')
```

zip(*i1, ..., iN*)

- 순회자 ***i1, ... , iN***을 사용하여 튜플 순회자를 반환한다.

In [ ]:

```
fruits = ['사과', '바나나', '수박', '오렌지']
stock = [10, 30, 20, 40]

for i in zip(fruits, stock):
    print(i)
```

In [ ]:

```
for f, n in zip(fruits, stock):
    print(f'{f}는 {n}개 남았습니다')
```