시퀀스형 자료구조

작성일시	@2022년 7월 18일 오후 2:14
내용	문법 자료구조 파이썬
주차	1주차
자료	

List

여러 개의 값을 **순서가 있는 구조**로 저장하고 싶을 때 사용

list의 생성과 접근

- 리스트는 대괄호([ ]) 혹은 list()를 통해 생성
  - 파이썬에서는 어떠한 자료형이든 넣을 수 있으며 리스트 안에 리스트를 넣을 수도 있음

```
#list_name = [var1, var2, var3, ...]
list_a = []
list_b = [1, 2, 3]
list_c = ['I', 'Love' 'You']
list_d = [1, 2, 3, 'Python', ['list' 'in' 'list']]
```

  - 생성된 이후에 내용 변경 가능 → mutable
  - 유연성 때문에 파이썬에서 가장 흔하게 사용
- 순서가 있는 시퀀스로 **인덱스**를 통해 접근 가능
  - 값에 대한 접근은 list[i]

```
my_list = []
another_list = list()
print(type(my_list)) # <class 'list'>
print(type(another_list)) # <class 'list'>

location = ['서울', '대전', '구미', '광주', '부울경']
print(location) # ['서울', '대전', '구미', '광주', '부울경']
print(type(location)) # <class 'list'>
print(location[0]) # 서울
```

```
location[0] = '양양'
print(location) # ['양양', '대전', '구미', '광주', '부울경']
```

Tuple

여러 개의 값을 순서가 있는 구조로 저장하고 싶을 때

- 생성 후 담고 있는 값 **변경 불가**
- 항상 소괄호 형태로 사용

Tuple의 생성과 접근

- 소괄호 () 혹은 tuple()을 통해 생성
- 튜플은 수정 불가능한(immutable) 시퀀스로 인덱스로 접근 가능
  - 값에 대한 접근은 tuple[i]
- 튜플 생성 시 주의 사항
  - 단일 항목
    - 하나의 항목으로 구성된 튜플은 생성 시 값 뒤에 쉼표를 붙여야됨
  - 복수 항목
    - 마지막 항목에 붙은 쉼표는 없어도 되지만 , 넣는 것을 권장(Trailing comma)

```
a = 1,
print(a) #(1,)

tuple_a = (1,)
print(tuple_a) # (1,)

b = 1, 2, 3,
print(b) #(1, 2, 3,)

tuple_b = (1, 2, 3,)
print(tuple_b) # (1, 2, 3,)
```

Tuple 대입 (Tuple assignment)

- 튜플 대입이란?
  - 우변의 값을 좌변의 변수에 할당하는 과정
- 튜플은 일반적으로 파이썬 내부에서 활용

- 추후 함수에서 복수의 값을 반환할 때 활용

```
x, y = 1, 2
print(x, y) = 1, 2

#실제로는 tuple로 처리
x, y = (1, 2)
print(x, y) # 1, 2
```

Range

숫자의 시퀀스를 나타내기 위해 사용

주로 반복문과 함께 사용됨

```
print(range(4)) # range(0,4)

#담겨 있는 숫자를 확인하기 위해서 리스트로 형변환(실제 활용시에는 필요x)
print(list(range(4))) # [0, 1, 2, 3]
print(type(range(4))) # <class 'range'>
```

Range의 사용 방법

- 기본형: range(n)
  - 0부터 n - 1 까지의 숫자 시퀀스
- 범위 지정: range(n, m)
  - n부터 m -1 까지의 숫자 시퀀스
- 범위 및 스텝 지정: range(n, m, s)
  - n부터 m -1 까지 s만큼 증가시키며 숫자의 시퀀스
  - s를 음수로 지정할 경우 **역순**

슬라이싱 연산자

시퀀스를 특정 단위로 슬라이싱

- 인덱스와 콜론을 사용하여 문자열의 특정부분만 잘라낼 수 있음

- 슬라이싱을 이용하여 문자열을 나타낼 때 콜론을 기준으로 **앞 인덱스에 해당하는 문자는 포함** 되지만 **뒤 인덱스에 해당 문자는 미포함**

```
# 리스트 ([1:4]에서 1은 포함 4는 미포함)
print([1, 2, 3, 5][1:4]) # [2, 3, 5]

# 튜플
print((1, 2, 3,)[:2]) # (1, 2)

# range
print(range(10)[5:8]) # range(5, 8)

# 문자열
print('abcd'[2:4]) # cd
```

시퀀스를 k 간격으로 슬라이싱

```
# 리스트 ([1:4]에서 1은 포함 4는 미포함)
print([1, 2, 3, 5][0:4:2]) # [1, 3]

# 튜플
print((1, 2, 3, 5)[0:4:2]) # (1, 3)

# range
print(range(10)[1:5:3]) # range(1, 5, 3)

# 문자열
print('abcdefg'[1:3:2]) # b
```