

11/11 월요일 (파이썬 스터디)

오늘 스터디 목표

1. 파이썬의 자료형 이해

(참고자료: "Do it! 점프 투 파이썬" 교재의 2장 파이썬 프로그래밍의 기초, 자료형)

2. 프로그래머스 최빈값 문제 리뷰

1. 파이썬의 자료형 이해

파이썬의 자료형(data type): 변수가 저장하는 값의 종류

"파이썬의 자료형"을 알아야 하는 이유: 자료형에 따라 데이터를 처리하는 방식이 달라지기 때문

<파이썬의 (기본적인)자료형 종류>

1. 숫자(int, float): 정수, 실수를 나타내는 자료형으로, 계산, 측정 값 저장 등에 사용됩니다.
2. 문자열(str): 문자나 단어, 문장을 저장하는 자료형으로, 텍스트 데이터 저장과 처리에 사용됩니다.
3. 불(bool): 참(True), 거짓(False)을 나타내는 자료형으로, 조건을 검사하거나 결과를 반환할 때 사용합니다.
4. 리스트(list)
5. 튜플(tuple)
6. 딕셔너리(dict)
7. 집합(set)

<자료형 상세 설명>

1. 리스트 (list)

- **정의:** 여러 개의 데이터를 순서대로 저장하는 자료형
- **사용 목적:** 순서가 중요한 데이터의 집합을 관리할 때 사용합니다.
- **구조:** 대괄호 `[]` 로 요소를 감싸고, 요소들은 쉼표 `,` 로 구분합니다.

```
my_list = [1, 2, 3, 'a', 'b']
```

- **특징:**
 - **순서 존재:** 요소들은 순서를 가지며, 인덱싱과 슬라이싱이 가능합니다.
 - **변경 가능:** 요소의 추가, 삭제, 수정이 가능합니다.
 - **중복 허용:** 동일한 값을 가진 요소를 여러 개 가질 수 있습니다.

2. 튜플 (tuple)

- **정의:** 리스트와 비슷하지만 한번 만들면 수정할 수 없는 자료형
- **사용 목적:** 변경되지 않아야 하는 데이터를 저장할 때 사용
- **구조:** 소괄호 `()` 로 요소를 감싸고, 요소들은 쉼표 `,` 로 구분합니다.

```
my_tuple = (1, 2, 3, 'a', 'b')
```

- **특징:**
 - **순서 존재:** 리스트와 마찬가지로 요소들의 순서가 있으며, 인덱싱과 슬라이싱이 가능합니다.
 - **변경 불가능:** 한 번 생성된 튜플은 수정할 수 없습니다.
 - **중복 허용:** 리스트와 마찬가지로 동일한 값을 가진 요소를 여러 개 가질 수 있습니다.

3. 딕셔너리 (dict)

- **정의:** 키(key)와 값(value)을 쌍으로 저장하는 자료형
- **사용 목적:** 키를 통해 값에 빠르게 접근하고 데이터를 관리할 때 사용합니다.
- **구조:** 중괄호 `{ }` 로 감싸고, 키와 값은 콜론 `:` 으로 구분하며, 각 쌍은 쉼표 `,` 로 구분합니다.

```
my_dict = {'name': 'Alice', 'age': 25}
```

- **특징:**
 - **순서 존재:** Python 3.7부터 삽입된 순서를 유지합니다.
 - **변경 가능:** 키-값 쌍의 추가, 삭제, 수정이 가능합니다.
 - **키의 중복 불가:** 키는 유일해야 하며, 중복된 키를 사용하면 마지막 값으로 덮어씌워집니다.

4. 집합 (set)

- **정의:** 중복되지 않는 데이터를 모아둔 자료형
- **구조:** 중괄호 `{ }` 로 감싸거나 `set()` 함수를 사용합니다.

```
my_set = {1, 2, 3}
```

```
# set() 함수를 사용하여 집합 생성
char_set = set("hello")
# {'h', 'e', 'l', 'o'} 중복이 제거된 상태
```

- **특징:**
 - **순서 없음:** 요소들의 순서가 정해져 있지 않으며, 인덱싱과 슬라이싱이 불가능합니다.
추가 설명: 인덱싱과 슬라이싱은 특정 위치에 있는 값을 index를 통해 접근하는 방법으로, 순서가 있는 데이터형(예: 리스트)에서만 사용 가능합니다. 집합(set)은 순서가 없기 때문에 이러한 방식으로 접근할 수 없습니다.
 - **변경 가능:** 요소의 추가, 삭제가 가능합니다.
 - **중복 불가:** 동일한 값을 가진 요소는 하나만 저장됩니다.

추가로 헛갈릴 만한 개념들

1. 배열 (array)

- **정의:** 같은 종류의 데이터를 메모리를 적게 사용하여 저장하는 자료형
- **Python에서의 사용:** 내장 모듈인 `array` 나 외부 라이브러리인 `numpy` 의 `array` 를 사용합니다.
(간단한 배열 기능만 필요하면 Python에 내장된 기본 배열 모듈인 `array` 사용)
(대량의 수치 연산에는 외부 라이브러리인 `numpy` 가 더 적합)

```
# 내장 array 모듈 사용 예시
import array

# 'i'는 4바이트 정수형을 의미
# 'i'는 배열 my_array의 모든 요소가 4바이트 정수형이어야 한다는 것을 의미
my_array = array.array('i', [1, 2, 3])
print(my_array) # array('i', [1, 2, 3])
```

```
# numpy 라이브러리 사용 예시
import numpy as np
my_numpy_array = np.array([1, 2, 3])
print(my_numpy_array) # [1 2 3]
```

- **특징:**

- **동일한 자료형:** 모든 요소는 동일한 자료형이어야 합니다.

부연 설명: 배열은 메모리 공간을 효율적으로 사용하기 위해 만들어졌습니다. 배열에서는 각 요소가 동일한 크기로 저장되므로, 컴퓨터가 일정한 간격으로 데이터를 빠르게 읽을 수 있습니다. 만약 요소의 자료형이 다르다면 크기가 달라져 배열 구조가 깨지고 메모리 접근 속도가 느려질 수 있습니다.

- **메모리 효율성:** 리스트보다 메모리 사용이 효율적입니다.

부연 설명: 리스트는 다양한 자료형을 저장할 수 있어 각 요소마다 추가적인 메모리 정보가 필요합니다. 반면 배열은 동일한 자료형만 저장하므로 메모리 관리가 단순해지고 불필요한 메모리 낭비를 줄일 수 있습니다.

- **사용 목적:** 대량의 수치 데이터를 효율적으로 저장하고 계산할 때 사용합니다.

2. 문자열 (string)

- **정의:** 글자들을 순서대로 저장한 자료형
- **사용 목적:** 텍스트 데이터를 저장하고 처리할 때 사용합니다.
- **구조:** 작은따옴표 `' '` 또는 큰따옴표 `" "`로 감쌉니다.

```
my_string = "Hello, World!"
```

- **특징:**

- **순서 유지:** 인덱싱과 슬라이싱이 가능합니다.

- **변경 불가능:** 기존 문자열 객체 자체를 **직접 변경할 수 없다.**(새로운 문자열을 만들거나 조합하여 수정할 수 있지만, 기존 문자열의 내용을 직접 변경하는 것은 불가능합니다.)

```
my_string = "Hello"
my_string[0] = "h" # 오류 발생! 문자열의 특정 문자 자체는 변경 불가능
```

자료형 비교 표

특성	리스트 (list)	튜플 (tuple)	딕셔너리 (dict)	집합 (set)	배열 (array)	문자열 (string)
선언 방법	<code>[]</code>	<code>()</code>	<code>{키:값}</code>	<code>{ }</code>	<code>array()</code>	<code>' '</code> 또는 <code>" "</code>
순서 있음?	예	예	예 (<i>Python 3.7 부터</i>)	아니오	예	예
수정 가능?	예	아니오	예	예	예	아니오
중복 허용?	예	예	키: 아니오	아니오	예	예

			값: 예			
인덱스로 접근?	예	예	아니오(키로 접근)	아니오	예	예
데이터 타입	숫자, 문자열, 리스트 등 모든 자료형	숫자, 문자열, 리스트 등 모든 자료형	키: 숫자, 문자열, 튜플만 가능 값: 모든 자료형 가능	숫자, 문자열, 튜플만 가능	같은 종류의 숫자만 가능	문자만 가능
주요 사용 목적	여러 데이터를 순서대로 저장	변경되면 안 되는 여러 데이터 저장	키를 통한 빠른 데이터 접근	중복 제거 및 집합 연산	효율적인 수치 계산	텍스트 데이터 처리

OX 퀴즈

1. 리스트는 변경 불가능한(immutable) 자료형이다.
2. 튜플은 요소의 순서가 없다.
3. 딕셔너리의 키는 변경 가능해야 한다.
4. 집합(set)은 중복된 요소를 허용하지 않는다.
5. 문자열은 변경 가능한(mutable) 자료형이다.
6. 배열(array)은 리스트와 달리 모든 요소가 동일한 자료형이어야 한다.
7. 딕셔너리에서 키를 사용하여 값을 빠르게 접근할 수 있다.
8. 리스트와 튜플 모두 슬라이싱(slicing)이 가능하다.
9. 집합은 요소의 순서를 유지한다.
10. 리스트는 딕셔너리의 키로 사용할 수 있다.