# 리스트

리스트를 만들 때는 위에서 보는 것과 같이 대괄호([ ])로 감싸 주고 각 요소 값들은 쉼표(,)로 구분해 준다.

리스트명 = [요소1, 요소2, 요소3, ...]

여러 가지 리스트의 생김새를 살펴보면 다음과 같다.

- 리스트는 아무것도 포함하지 않는, 비어 있는 리스트([ ])일 수도 있고

- 숫자를 요소 값으로 가질 수도 있고

- 문자열을 요소 값으로 가질 수도 있다.

- 숫자와 문자열을 함께 요소 값으로 가질 수도

- 리스트 자체를 요소 값으로 가질 수도 있다.

=> 즉, 리스트 안에는 어떠한 자료 형도 포함시킬 수 있다.

## 리스트에 요소 추가 (append)

append(x)는 리스트의 맨 마지막에 x를 추가시키는 함수이다.

## 리스트 정렬 (sort): 리스트의 요소를 순서대로 정렬

## 리스트 뒤집기(reverse): reverse 함수는 리스트를 역순으로 뒤집어 준다.

## 위치 반환(index): 위치 값 리턴

## 리스트에 요소 삽입(insert)

insert(a, b) 리스트의 a번째 위치에 b를 삽입

ex) a= [1,2,3]

a.insert(0,4)

a #[4,1,2,3]

## 리스트 요소 제거(remove)

remove(x): 리스트에서 첫 번째로 나오는 x를 삭제하는 함수

## 리스트 요소 끄집어내기(pop)

pop() 리스트의 맨 마지막 요소를 돌려 주고 그 요소를 삭제하는 함수

# 튜플

## 리스트와의 차이점

|  |  |
| --- | --- |
| 리스트 | 튜플 |
| [, ] | (, ) |
| 값의 생성, 삭제, 수정이 가능 | 불가능 |

튜플과 리스트의 가장 큰 차이는 값을 변경시킬 수 있는가 없는가이다.

즉, 리스트의 항목 값은 변경이 가능하고 튜플의 항목 값은 변경이 불가능하다.

# 딕셔너리

딕셔너리란?

리스트나 튜플처럼 순차적으로 해당 요소 값을 구하지 않고, key를 통해 value를 얻는다.

즉, baseball이라는 단어의 뜻을 찾기 위해 사전의 내용을 순차적으로 모두 검색하는 것이 아니라 baseball이라는 단어가 있는 곳만 펼쳐 보는 것이다.

## 기본적인 딕셔너리

{Key1:Value1, Key2:Value2, Key3:Value3, ...}

\*\*key에는 변하지 않는 값을 사용하고, value에는 변하는 값과 변하지 않는 값 모두 사용 가능.

## 딕셔너리에서 Key를 사용해 Value 얻기.

|  |
| --- |
| >>> grade = {'pey': 10, 'julliet': 99}  >>> grade['pey']  10  >>> grade['julliet']  99 |

리스트나 튜플, 문자열은 요소 값을 얻어내고자 할 때 인덱싱이나 슬라이싱 기법 중 하나를 이용했다.

하지만 딕셔너리는 key를 사용해서 value를 얻어내는 방법 뿐이다.

ex) grade[‘pey’]

## 딕셔너리 관련 함수들

-key 리스트 만들기(keys)

|  |
| --- |
| >>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}  >>> a.keys()  dict\_keys(['name', 'phone', 'birth']) |

a.keys()는 딕셔너리 a의 key만을 모아서 dict\_keys라는 객체를 리턴

-value 리스트 만들기(values)

|  |
| --- |
| >>> a.values()  dict\_values(['pey', '0119993323', '1118']) |

-key, value 쌍 얻기(items())

-key:value 쌍 모두 지우기(clear())

-key로 value 얻기 (get)

|  |
| --- |
| >>> a = {'name':'pey', 'phone':'0119993323', 'birth': '1118'}  >>> a.get('name')  'pey'  >>> a.get('phone')  '0119993323' |

-해당 key가 딕셔너리 안에 있는지 조사하기(in)

|  |
| --- |
| >>> a = {'name':'pey', 'phone':'0119993323', 'birth': '1118'}  >>> 'name' **in** a  **True**  >>> 'email' **in** a  **False** |

# 집합 자료형

## 특징

- 중복을 허용하지 않는다.

- 순서가 없다(unordered)

\*\*중복을 허용하지 않는 set의 특징은 자료형의 중복을 제거하기 위한 필터 역할로도 사용

## 관련 함수들

값 1개 추가(add)

값 여러 개 추가(update)

특정 값 제거(remove)