

# Lecture #08 | 집합(set), 사전(dict)

SE213 프로그래밍 (2018)

# 오늘 다룰 내용

---

- 자료구조
  - 집합(set)
  - 사전(dict)

# 집합(set)

---

- 집합: 순서가 없고 중복이 없는 원소들을 포함 (수학에서 집합과 유사)
  - 원소의 자료형: 불변하는(immutable) 형(부울, 정수, 실수, 튜플, 문자열)만 가능
    - 일반적으로 문자열 혹은 정수가 많이 사용됨
- 정의
  - {}를 이용하고, 원소들을 ,로 구분하여 정의
  - 원소들을 포함하는 리스트, 튜플 등을 `set()` 함수의 인자로 하여 형변환
  - 단, 빈 집합(empty set)을 정의하고 싶을 때는 원소 없이 `set()` 함수만 사용함
- 출력: {}를 사용

## 예시: 집합의 연산자 (1/2)

```
s1 = {42, 1024, 23}
print(s1)
s2 = set() # empty set
print(s2)
s2.add(6)
print(s2)
s2.add(23)
print(s2)
print(42 in s1)
print(42 in s2)
print(len(s1)) # number of elements in s1
print(len(s2)) # number of elements in s2
```

```
{1024, 42, 23}
set()
{6}
{6, 23}
True
False
3
2
```

## 예시: 집합의 연산자 (2/2)

```
print(s1 | s2) # union
print(s1.union(s2))
print(s1 - s2) # difference
print(s1.difference(s2))
print(s2 - s1) # difference
print(s2.difference(s1))
print(s1 & s2) # intersection
print(s1.intersection(s2))
print(s1 ^ s2) # symmetric difference
print(s1.symmetric_difference(s2))
s3 = s1.copy()
print(s3)
s3.clear()
print(s3)
```

```
{1024, 6, 42, 23}
{1024, 6, 42, 23}
{1024, 42}
{1024, 42}
{6}
{6}
{23}
{23}
{1024, 42, 6}
{1024, 42, 6}
{1024, 42, 23}
set()
```

# 사전(dict)

---

- 리스트, 튜플
  - 여러 아이템을 하나의 변수 이름에 저장
  - 숫자를 이용한 인덱스로 아이템에 접근 가능
- 사전(dict): '키(key)'와 '값(value)'의 쌍들로 구성된 mutable mapping
  - 키
    - 불변하는 타입(부울, 정수, 실수, 튜플, 문자열)을 사용
    - 일반적으로 문자열 혹은 정수가 많이 사용됨
  - 값: 파이썬의 어떠한 데이터 형태(리스트, 튜플, 사전 등도 포함)도 가능
- 참고: 다른 언어에서는 associative memory, associative array, hash, hashmap 등으로 불림

# 예시: 사전의 정의와 사용

```
bts_position = {  
    '랩몬스터': '리더, 메인래퍼',  
    '진': '서브보컬',  
    '슈가': '리드래퍼',  
    '제이홉': '서브래퍼, 메인댄서',  
    '지민': '리드보컬, 리드댄서',  
    '뷔': '서브보컬',  
    '정국': '메인보컬, 서브래퍼, 리드댄서'  
}  
  
print(bts_position)  
print(bts_position['뷔'])  
print(bts_position['정국'])
```

```
{'랩몬스터': '리더, 메인래퍼', '진': '서브보컬', '슈가': '리드래퍼', '제이홉': '서브래퍼, 메인댄서', '지민': '리드보컬, 리드댄서', '뷔': '서브보컬', '정국': '메인보컬, 서브래퍼, 리드댄서'}
```

서브보컬

메인보컬, 서브래퍼, 리드댄서

Global frame

bts\_position

dict



"랩몬스터"	"리더, 메인래퍼"
"진"	"서브보컬"
"슈가"	"리드래퍼"
"제이홉"	"서브래퍼, 메인댄서"
"지민"	"리드보컬, 리드댄서"
"뷔"	"서브보컬"
"정국"	"메인보컬, 서브래퍼, 리드댄서"

# 항목 추가하기

- 새로운 키/값을 이용하여 원소를 추가할 수 있음
  - 리스트와 다르게 초기화되지 않은 키(리스트의 인덱스에 해당)를 사용할 수 있음
- 참고: 값을 읽을 때는 미리 정의된 키만 사용할 수 있음
  - 정의되지 않은 키를 사용하면 `KeyError`가 발생

```
numbers = {'pi': 3.14, 'e': 2.718}
numbers['golden_ratio'] = 1.618 # add a new item
print(numbers)
print(numbers['answer']) # KeyError: 'answer'
```

```
{'pi': 3.14, 'e': 2.718, 'golden_ratio': 1.618}
```



## 사전으로 형변환: dict() 함수 사용

- 두 개의 원소를 가진 시퀀스를 사전으로 형변환 가능함
  - 각 시퀀스의 첫 번째 항목은 키(key), 두 번째 항목을 값(value)으로 사용됨

```
t1 = [['answer', 42], ['pi', 3.14], ['e', 2.718]]
d1 = dict(t1)
print(d1)

t2 = [['answer', 42], ['pi', 3.14], ['e', 2.718]]
d2 = dict(t2)
print(d2)
```

```
{'answer': 42, 'pi': 3.14, 'e': 2.718}
{'answer': 42, 'pi': 3.14, 'e': 2.718}
```

# 키 멤버십 테스트: in, not in

- 특정한 키가 사전에 속해있는지 확인 (참고: 키만 확인하고 값은 확인하지 않음)

```
bts_position = {  
    '랩몬스터': '리더, 메인래퍼',  
    '진': '서브보컬',  
    '슈가': '리드래퍼',  
    '제이홉': '서브래퍼, 메인댄서',  
    '지민': '리드보컬, 리드댄서',  
    '뷔': '서브보컬',  
    '정국': '메인보컬, 서브래퍼, 리드댄서'}  
print('진' in bts_position)  
print('슈가' not in bts_position)  
print('서브보컬' in bts_position)
```

```
True  
False  
False
```

# keys(), values(), items()

---

- 각각 키, 값, 키와 값의 쌍을 튜플로 반환함
  - 인덱싱이 필요한 경우, list() 함수를 이용하여 리스트로 형변환

```
numbers = {'pi': 3.14, 'e': 2.718, 'gr': 1.618}
print(numbers.keys())
print(numbers.values())
print(numbers.items())
print(list(numbers.items())[2])
```

```
dict_keys(['pi', 'e', 'gr'])
dict_values([3.14, 2.718, 1.618])
dict_items([('pi', 3.14), ('e', 2.718), ('gr', 1.618)])
('gr', 1.618)
```

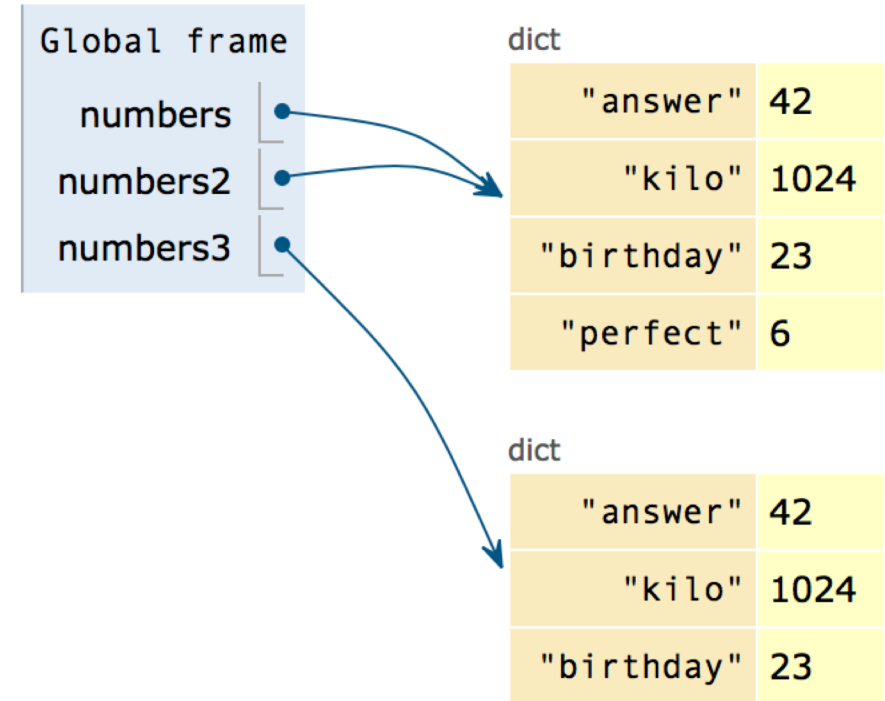
## 예시: keys(), values(), items()를 for문과 사용

```
bts_position = {  
    '랩몬스터': '리더, 메인래퍼',  
    '진': '서브보컬',  
    '슈가': '리드래퍼',  
    '제이홉': '서브래퍼, 메인댄서',  
    '지민': '리드보컬, 리드댄서',  
    '뷔': '서브보컬',  
    '정국': '메인보컬, 서브래퍼, 리드댄서'}  
for nickname in bts_position.keys():  
    print(nickname)  
for nickname, position in bts_position.items():  
    print(nickname, ': ', position)
```

```
랩몬스터  
진  
슈가  
제이홉  
지민  
뷔  
정국  
랩몬스터 : 리더, 메인래퍼  
진 : 서브보컬  
슈가 : 리드래퍼  
제이홉 : 서브래퍼, 메인댄서  
지민 : 리드보컬, 리드댄서  
뷔 : 서브보컬  
정국 : 메인보컬, 서브래퍼, 리드댄서
```

## 예시: 사전의 복사, copy() 함수

```
numbers = {'answer': 42, 'kilo': 1024,
           'birthday': 23}
numbers2 = numbers
numbers3 = numbers.copy()
numbers2['perfect'] = 6
print(numbers)
print(numbers2)
print(numbers3)
```



```
{'answer': 42, 'kilo': 1024, 'birthday': 23, 'perfect': 6}
{'answer': 42, 'kilo': 1024, 'birthday': 23, 'perfect': 6}
{'answer': 42, 'kilo': 1024, 'birthday': 23}
```

## 항목 삭제하기: del, clear()

---

- `del d[key]`: Remove `d[key]` from `d`. Raises a `KeyError` if `key` is not in the map.
- `dict.clear()`: Remove all items from the dictionary.

```
numbers = {'answer': 42, 'kilo': 1024, 'birthday': 23}
print(numbers)
del numbers['kilo']
print(numbers)
numbers.clear()
print(numbers)
```

```
{'answer': 42, 'kilo': 1024, 'birthday': 23}
{'answer': 42, 'birthday': 23}
{}
```

## 항목 얻기: get()

---

- `dict.get(key[, default])`: Return the value for `key` if `key` is in the dictionary, else `default`. If `default` is not given, it defaults to `None`, so that this method never raises a `KeyError`.

```
numbers = {'answer': 42, 'kilo': 1024, 'birthday': 23}
numbers['perfect'] = 6 # new item is added
#print(numbers['prime']) # KeyError: 'prime'
print(numbers.get('answer', 0))
print(numbers.get('prime', 0))
```

42

0

## 예시: sequence의 아이템 수 세기

```
def count_items(sequence):  
    count = {}  
    for item in sequence:  
        if item in count:  
            count[item] += 1  
        else:  
            count[item] = 1  
    return count
```

```
t = [1, 2, 3, 1, 2, -1, 4]  
print(count_items(t))
```

```
{1: 2, 2: 2, 3: 1, -1: 1, 4: 1}
```



## 예시: sequence의 아이템 수 세기 - get() 함수 이용

```
def count_items_short(sequence):  
    count = {}  
    for item in sequence:  
        count[item] = count.get(item, 0) + 1  
    return count
```

```
t = [1, 2, 3, 1, 2, -1, 4]  
print(count_items_short(t))
```

```
{1: 2, 2: 2, 3: 1, -1: 1, 4: 1}
```

# 리스트, 튜플, 사전, 집합 비교

	리스트(list)	튜플(tuple)	사전(dict)	집합(set)
정의	<code>t = [1, 2, 3]</code>	<code>t = (1, 2, 3)</code> <code>t = 1, 2, 3</code>	<code>t = {'1': 3, '2': 4, '3': -1}</code>	<code>t = {1, 2, 3}</code>
인덱스 방식	정수 인덱스	정수 인덱스	(immutable인) 키 예: 정수, 문자열, 튜플	n/a
인덱스 예시	<code>t[2]</code>	<code>t[2]</code>	<code>t['2']</code>	n/a
변경 가능	가능	불가능	가능	가능



---

ANY QUESTIONS?