

Lesson 7

Collection 자료형
딕셔너리(Dictionary)



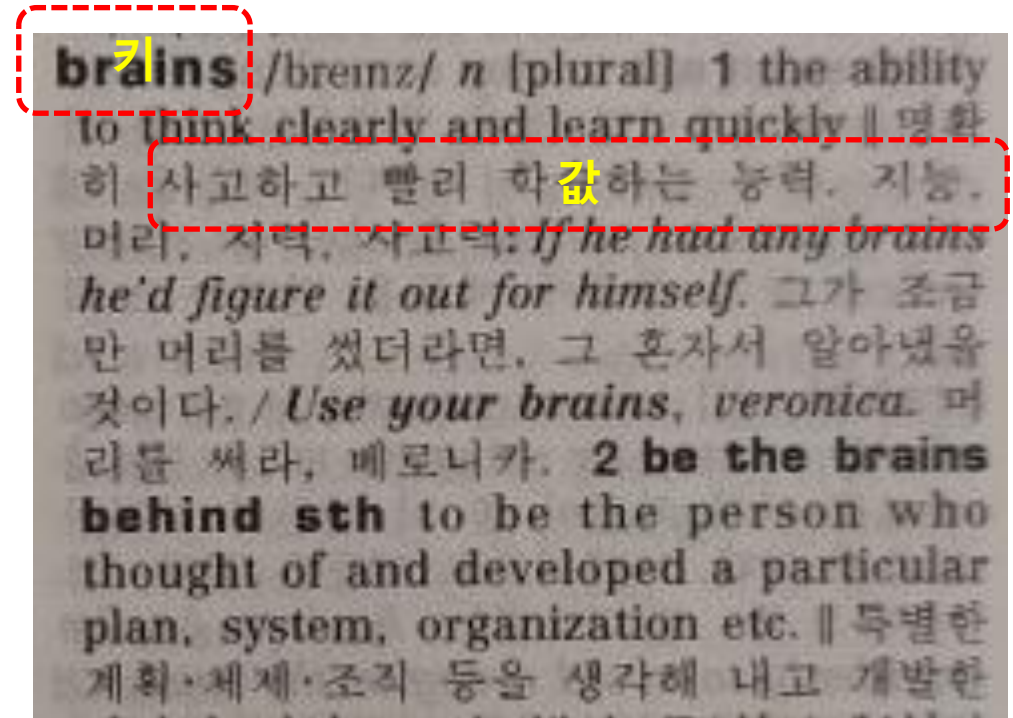
목차

1. 딕셔너리

- 딕셔너리 생성
- 딕셔너리 접근
- 딕셔너리 추가, 수정, 삭제
- 딕셔너리 관련 함수

딕셔너리(dict)

- 키(key) + 값(value)로 구성된 자료형 -> {키 : 값}
- 순서X, 키 중복X, 수정O, 삭제O, 추가O



■ 변수={키1:값1, 키2:값2,...}

■ 키(key)는 중복 불가.

■ 키(Key)에 리스트는 쓸 수 없고 튜플은 쓸 수 있다.

■ 값(Value)에는 어떤 값이든 사용할 수 있다.



unhashable 객체는 키로 사용 불가.

코드

```
d1={1:10,2:20,3:30,4:40}
```

```
d2={"name":["kim","lee","park"],"score):(100,90,50)}
```

```
d3=dict(  
    name="kim",  
    age=20,  
    address="seoul"  
)
```

딕셔너리 접근 : [키] or get() 함수

코드

```
d={1:"one","cities":("seoul","jeju"),'a':"apple",'b':[5,6,7]}  
print(d[1])                # 키가 1인 요소의 값. one  
print(d['a'])               # 키가 a인 요소의 값. apple  
print(d["cities"])          # 키가 cities인 요소의 값. ("seoul","jeju")  
print(d.get("cities"))      # 키가 cities인 요소의 값. ("seoul","jeju")
```



get()는 찾는 키가 없으면 None 처리.

딕셔너리 수정 : [키]=값 or update()

코드

```
d={"apple":"사과","grape":"포도"}
```

```
d["banana"]="버네너"
```

```
print(d)    # {"apple": "사과", "grape": "포도", "banana": "버네너"}
```

```
d.update(banana="빠나나", orange="오렌지")
```

```
print(d)    # {"apple": "사과", "grape": "포도", "banana": "빠나나", "orange": "오렌지"}
```



해당 키가 없으면 추가되며 있으면 덮어쓰기됨.

딕셔너리 추가 : [키]=값, 삭제 : pop(키) or popitem()

코드

```
d={"apple":"사과","grape":"포도"}
```

```
# 요소 추가
```

```
d["banana"]="바나나"
```

```
print(d)    # {"apple": "사과", "grape": "포도", "banana": "바나나"}
```

```
# 요소 삭제
```

```
d.pop("banana")
```

```
print(d)    # {"apple": "사과", "grape": "포도"}
```

```
d.popitem()
```

```
print(d)    # {"apple": "사과"}
```

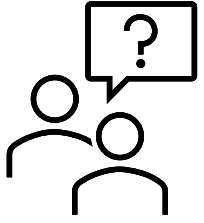


popitem(): 마지막 요소를 삭제한다.

- key 리스트 얻기: `keys()`
- value 리스트 얻기: `values()`
- key, value 얻기: `items()`

코드

```
d = {  
    "name": "kim",  
    "phone": "0109993323",  
    "birth": "1118"  
}  
  
print(d.keys())           # dict_keys(["name", "phone", "birth"])  
print(d.values())         # dict_values(["kim", "0109993323", "1118"])  
print(d.items())          # dict_items([("name", "kim"), ("phone", "0109993323"), ("birth",  
"1118")])  
print(list(d.keys()))      # ["name", "phone", "birth"]
```



1. 딕셔너리는 순서가 (있다 없다).
2. 딕셔너리는 요소를 추가할 수 (있다 없다).
3. 딕셔너리는 요소를 수정할 수 (있다 없다).
4. 딕셔너리는 중복된 키를 가질 수 (있다 없다).
5. 딕셔너리는 생성하려면 ()를 사용하면 된다.
6. 빈 딕셔너리를 만들려면?

→ { } , dict()

도전!

[문제] 입력한 영어 단어 출력하기

```
words={"사과":"apple", "고양이":"cat", "아몬드":"almond", "접시":"dish"}
```

["사과", "고양이", "아몬드", "접시"]중 하나 입력: 사과
사과 --> apple

코드

```
words={"사과":"apple", "고양이":"cat", "아몬드":"almond", "접시":"dish"}

print(list(words.keys()), end=" ")
word=input("중 하나 입력: ")

print(f"{word} --> {words[word]}")
```

도전!

[문제] 입력한 음식과 궁합이 맞는 음식 출력 프로그램.

```
foods = {  
  "떡볶이": "오뎅",  
  "짜장면": "단무지",  
  "라면": "김치",  
  "피자": "피클",  
  "맥주": "땅콩",  
  "치킨": "치킨무",  
  "삼겹살": "상추"}  
}
```

["떡볶이", "짜장면", "라면", "피자", "맥주", "치킨", "삼겹살"]중 좋아하는 음식은? 피자
"피자"에는 "피클"입니다.

딕셔너리 구조 이해하기

```
data={'response': {'header': {'resultCode': '00', 'resultMsg': 'NORMAL SERVICE.'},
  'body': {'items': {'item': [{'거래금액': '82,500', '건축년도': '2008', '도로명': '세종대로23길'},
    {'거래금액': '130,000', '건축년도': '2004', '도로명': '경희궁2길'}]},
  'numOfRows': '10',
  'pageNo': '1',
  'totalCount': '49'}}
}
```

```
cctv={'fields':
  [
    {'id': '점포명'},
    {'id': '소재지도로명주소'}
  ],
  'records':
  [
    {'점포명': 'CU 도계사랑점', '소재지도로명주소': '경상남도 창원시 의창구 원이대로81번길 28'},
    {'점포명': '세븐일레븐(소답점)', '소재지도로명주소': '경상남도 창원시 의창구 의안로2번길 25'},
    {'점포명': 'GS25 봉곡성우점', '소재지도로명주소': '경상남도 창원시 의창구 지귀로 14'}
  ]
}
```

pprint 모듈로 딕셔너리 데이터를 보기 좋게 정렬하기

```
import pprint

real_estate = {'response': {'header': {'resultCode': '00', 'resultMsg': 'NORMAL SERVICE.'}, 'body': {'items': {'item': [{'거래금액': '82,500', '건축년도': '2008', '도로명': '세종대로23길'}, {'거래금액': '130,000', '건축년도': '2004', '도로명': '경희궁2길'}]}}, 'numOfRows': '10', 'pageNo': '1', 'totalCount': '49'}}}

pprint.pprint(real_estate)
```

실행 결과 ►

```
{'response': {'body': {'items': {'item': [{'거래금액': '82,500',
                                           '건축년도': '2008',
                                           '도로명': '세종대로23길'},
                                           {'거래금액': '130,000',
                                           '건축년도': '2004',
                                           '도로명': '경희궁2길'}]}},
               'numOfRows': '10',
               'pageNo': '1',
               'totalCount': '49'},
 'header': {'resultCode': '00', 'resultMsg': 'NORMAL SERVICE.'}}}
```

json 데이터 보기 좋게 정렬하는 사이트들

<https://tools.arantius.com/tabifier>

<https://jsoneditoronline.org/>

<https://jsonviewer.stack.hu/>

<https://beautifytools.com/xml-viewer-editor.php>

```

{
  "response": {
    "header": {
      "resultCode": "00",
      "resultMsg": "NORMAL SERVICE."
    },
    "body": {
      "items": {
        "item": [
          {
            "거래금액": "82,500",
            "건축년도": "2008",
            "도로명": "세종대로23길"
          },
          {
            "거래금액": "130,000",
            "건축년도": "2004",
            "도로명": "경희궁2길"
          }
        ]
      },
      "numOfRows": "10",
      "pageNo": "1",
      "totalCount": "49"
    }
  }
}

```

```

{
  "fields": [
    {
      "id": "점포명"
    },
    {
      "id": "소재지도로명주소"
    }
  ],
  "records": [
    {
      "점포명": "CU 도계사랑점",
      "소재지도로명주소": "경상남도 창원시 의창구 원이대로81번길 28"
    },
    {
      "점포명": "세븐일레븐(소답점)",
      "소재지도로명주소": "경상남도 창원시 의창구 의안로2번길 25"
    },
    {
      "점포명": "GS25 봉곡성우점",
      "소재지도로명주소": "경상남도 창원시 의창구 지귀로 14"
    }
  ]
}

```


도전!

[문제] 아래 **data** 변수에서 점포명만 추출해서 출력하세요.

```
data = {'fields':  
    [  
        {'id': '점포명'},  
        {'id': '소재지도로명주소'}  
    ],  
    'records':  
    [  
        {'점포명': 'CU 도계사랑점', '소재지도로명주소': '경상남도 창원시 의창구 원이대로81번길 28'},  
        {'점포명': '세븐일레븐(소답점)', '소재지도로명주소': '경상남도 창원시 의창구 의안로2번길 25'},  
        {'점포명': 'GS25 봉곡성우점', '소재지도로명주소': '경상남도 창원시 의창구 지귀로 14'}  
    ]  
}  
  
print(cctv['records'][0]['점포명'])  
print(cctv['records'][1]['점포명'])  
print(cctv['records'][2]['점포명'])
```

CU 도계사랑점
세븐일레븐(소답점)
GS25 봉곡성우점