딕셔너리의 이해

10주차_02

한 동 대 학 교
김경미 교수

사전형(Dictionary) 이란?

- **사전은 리스트와 비슷하지만**
  - 리스트에서는 index 값이 정수여야 함
  - 사전에서는 모든 종류의 데이터 형을 index로 사용 가능(user-defined indexes)
  - 정의할 때, '{'와 '}'를 사용

사전형 정의하기

- 정의하기

```
>>> days = { ‘Sun’:’Sunday’, ‘Mon’:’Monday’ , ‘Tue’:’Tuesday’,
‘Wed’:’Wednesday’, ‘Thu’:’Thursday’, ‘Fri’:’Friday’, ‘Sat’:’Saturday’}
```

- 첫번째 기술한 아이템이 찾을 때 사용하는 키(index)가 된다

```
>>> days[ ‘Sun’ ]
‘Sunday’
>>> days[ ‘Fri’ ]
‘Friday’
```

사전형 생성 후 자료 추가

- **비어있는 사전 생성**

  >>> d = { }

  - 함수 dict는 항목(item)없는 새로운 사전을 생성한다
    - 생성 후, 다음과 같이 아이템을 한 개씩 추가 한다

```
# create dictionary
>>> eng2sp = dict()
>>> print(eng2sp)
{}
>>> eng2sp['one'] = 'uno'
>>> print(eng2sp)
{'one': 'uno'}
```

사전형 연산

operator	Description
len()	사전형 변수에 저장된 아이템의 개수를 알려준다
k in dictionary_name	k 가 해당 사전형에 존재하면 True, 존재하지 않으면 False
Dictionary_name[k]	k 가 해당 사전형에 존재하면, 해당 아이템 값을 출력한다
Dictionary_name[k] = “item 값”	k 가 해당 사전형에 존재하지 않으면, 해당 아이템 index, 값이 추가되어 저장된다
Dictionary_name.pop(k)	k 가 해당 사전형에 존재하면, 삭제한다

사전형 연산, 개수와 아이템 확인

- **국가명과 국제전화 코드를 사전형으로 정의**

```
>>> country_code = {1:"미국", 20:"이집트", 30:"그리스", 39:"이태리", 81:"일본", 82:"한국"}
>>> len(country_code)
6

>>> country_code[30]
"그리스"

>>> country_code[82]
"한국"
```

사전형 연산, 자료 추가와 삭제

```
>>> 82 in country_code
True

>>> 60 in country_code
False

>>> country_code[60] = "말레지아"                    # append an item
>>> 60 in country_code
True

>>> country_code.pop(81)                            # delete an item
'일본'
>>> 81 in country_code
False
```

사전형 연산, 자료 읽기

```
# 사전형 city 값을 읽어내기
>>> city = {"New York City":8175133, "Los Angeles": 3792621, "Washington":632323,
"Chicago": 2695598, "Toronto":2615060, "Montreal":11854442, "Ottawa":883391,
"Boston":62600}
>>> city[“Toronto”]
2615060
>>> city[“Boston”]
62600

# 사전형 food 아이템 추가하기
>>> food = {"ham" : "yes", "egg" : "yes", "spam" : "no" }
>>> food
{'egg': 'yes', 'ham': 'yes', 'spam': 'no'}
>>> food[“spam”] = “yes”
>>> food
{'egg': 'yes', 'ham': 'yes', 'spam': 'yes'}
```

연습문제 1

- 아이템이 없는 사전형 birthdate를 정의한다
- 사용자에게 이름과 생일을 입력 받아서 사전형 birthdate에 추가한다
- 5명의 이름과 생일을 입력 받아서 추가한 후, 추가된 내용을 화면에 출력한다
- 특정한 한 사람의 이름을 입력 받아서, 생일을 화면에 출력한다

연습문제 1 코드

```
birthdate = { }
def add_birth(num):
    for i in range(num):
        input_name = input("이름: ")
        input_birth = int(input("생일: "))
        birthdate[input_name] = input_birth

add_birth(5)
print(birthdate)

name=input("생일을 찾고 싶은 사람의 이름을 입력하세요: ")
print(birthdate[name])
```

```
>>>
======== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/9주차_연습.py =
이름: kmkim
생일: 0224
이름: sjkang
생일: 0512
이름: kdhong
생일: 1212
이름: ychoi
생일: 0409
이름: hjkim
생일: 1103
{'kmkim': 224, 'sjkang': 512, 'kdhong': 1212, 'ychoi': 409, 'hjkim': 1103}
생일을 찾고 싶은 사람의 이름을 입력하세요: kmkim
224
```

연습문제 2

- 연습문제 1에서 생성한 dictionary 사용한다
- 입력 받은 사람의 자료를 삭제한다
- 삭제 후 in을 사용하여 삭제 여부를 확인한다

연습문제 2 코드

```
birthdate = { }
def add_birth(num):
    for i in range(num):
        input_name = input("이름: ")
        input_birth = int(input("생일: "))
        birthdate[input_name] = input_birth

add_birth(5)
print(birthdate)

name=input("생일을 삭제하고 싶은 사람의 이름을 입력하세요: ")
birthdate.pop(name)
print(name in birthdate)
print(birthdate)
```

```
생일을 삭제하고 싶은 사람의 이름을 입력하세요: kmkim
False
{'sjkang': 512, 'kdhong': 1212, 'ychoi': 409, 'hjkim': 1103}
>>>
```

Dictionary Method

Method	Description
d.items	사전형 d을 index-값 형식의 튜플로 구성된 리스트로 생성
d.get(k)	d[k] 와 같은 결과로 index k에 해당하는 값을 보여준다
d.keys(k)	사전형 d에서 index값만 찾아서 보여준다
d.pop(k)	사전형 d에서 index k인 아이템을 삭제한다
d.values(k)	사전형 d에서 index값을 제외하고, 값들만 찾아서 보여준다
d.update(d2)	사전형 d의 모든 저장된 내용에, 사전형 d2의 내용이 추가된다
dict(listname)	index-값 형식의 튜플로 구성된 리스트를 dictionary로 변환한다, d.items와 반대의 기능을 수행

Dictionary Methods, items

- **items라는 메소드**
  - index-값 형식의 튜플로 구성된 리스트를 생성

```
>>> d = {'a':0, 'b':1, 'c':2}                    #create dictionary
>>> t = d.items()
>>> t
[('a', 0), ('c', 2), ('b', 1)]                    # create a list contains
tuples
```

Dictionary Methods, get(k), keys()

.get(k) **.keys()**

```
>>> sp={'one':'uno', 'two':'dos', 'three':'tres'}
>>> sp.get('two')
'dos'
>>> sp['one']
'uno'
>>> sp.keys()
dict_keys(['one', 'two', 'three'])
```

요약

- 딕셔너리가 무엇인지 이해한다
- 딕셔너리를 정의하고 읽고, 자료 추가 삭제를 연습한다
- 딕셔너리 메소드 items(), get(), keys() 기능을 이해한다

감사합니다

10주차_02 딕셔너리의 이해