
리스트튜플딕셔너리 활용

10주차_03

한 동 대 학 교
김 경 미 교수

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 아이টে를 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, pop(k), values()

- **.pop(k)**

- **.values()**

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

Dictionary Methods, update(d2)

- 다른 딕셔너리의 내용을 추가한다

```
>>> sp={'one':'uno', 'two':'dos', 'three':'tres'}
>>> birthdate={'kmkim': 224, 'sjkang': 512, 'kdhong': 1212, 'ychoi': 409, 'hjkim': 1103}

>>> sp.update(birthdate)

>>> sp
{'one': 'uno', 'two': 'dos', 'three': 'tres', 'kmkim': 224, 'sjkang': 512, 'kdhong': 1212,
'ychoi': 409, 'hjkim': 1103}

>>> birthdate
{'kmkim': 224, 'sjkang': 512, 'kdhong': 1212, 'ychoi': 409, 'hjkim': 1103}
```

Dictionary Methods, dict

- **dict 메소드**

- index-값 형식의 튜플로 구성된 리스트를 dictionary로 변환

```
# 튜플로 구성된 리스트를 딕셔너리로 바꾸는 dict
>>> t = [('a', 0), ('c', 2), ('b', 1)]
>>> d = dict(t)
>>> d
{'a': 0, 'c': 2, 'b': 1}

>>> d = dict(zip('abc', range(3)))
>>> d
{'a': 0, 'b': 1, 'c': 2}
```

Dictionary 다수 값 정의

- Index는 하나이지만, 해당하는 값을 리스트로 구성하여 저장되는 사전형

```
>>> sp={'one':['uno', 1], 'two':['dos',2], 'three':['tres',3]}
```

```
>>> sp['one']
```

```
['uno', 1]
```

```
>>> sp.values()
```

```
dict_values(['uno', 1], ['dos', 2], ['tres', 3])
```

```
>>> sp.keys()
```

```
dict_keys(['one', 'two', 'three'])
```

```
>>> sp
```

```
{'one': ['uno', 1], 'two': ['dos', 2], 'three': ['tres', 3]}
```

연습문제 1

- 친구 이름과 전화번호로 구성된 사전형 phone을 구성한다
- 아이템 추가로 5명의 자료를 입력, 이 때 입력은 함수로 구성한다
- 입력된 자료의 index만 모두 출력한다
- 입력된 자료의 값들만 모두 출력한다

연습문제 1 코드

```
phone = {}
```

```
def add_phonenumber(n):  
    for i in range(n):  
        print(i+1)  
        index = input('이름: ')  
        phonenumber = int(input('전화번호: '))  
        phone[index] = phonenumber
```

```
add_phonenumber(5)
```

```
print(phone.keys())  
print(phone.values())
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/9주차_연습  
1  
이름: kmkim  
전화번호: 0100000000  
2  
이름: sjkang  
전화번호: 0101111111  
3  
이름: kdhong  
전화번호: 0102222222  
4  
이름: ychoi  
전화번호: 0103333333  
5  
이름: hjkim  
전화번호: 0104444444  
dict_keys(['kmkim', 'sjkang', 'kdhong', 'ychoi', 'hjkim'])  
dict_values([100000000, 101111111, 102222222, 103333333, 104444444])  
>>> |
```


연습문제 2

- **교사명과 담당교과목명을 사전형으로 구성한다**
 - '김경미' '수학', '과학'
 - '최영희' '영어', '수학'
 - '강동원' '영어'
 - '정필수' '사회', '역사'
 - '박희수' '국어'
 - '이승철' '수학', '과학'
- **교사명을 입력하면 담당교과목명을 출력한다**

연습문제 2 코드

```
Subjects = {'김경미':['수학','과학'],'최영희':['영어','수학'],'강동원':['영어','정필수':['사회','역사'],'박희수':['국어'],'이승철':['수학','과학']}
```

```
lec = input('담당교사명을 입력하시오: ')  
print(Subjects[lec])
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/  
담당교사명을 입력하시오: 강동원  
영어  
>>>  
===== RESTART: E:/1_Works/2017Work/  
담당교사명을 입력하시오: 이승철  
['수학', '과학']  
>>> |
```

연습문제 3

- 연습문제 3과 같은 딕셔너리를 사용한다
- 교사명과 담당교과목명을 사전형으로 구성한다
 - '김경미' '수학', '과학'
 - '최영희' '영어', '수학'
 - '강동원' '영어'
 - '정필수' '사회', '역사'
 - '박희수' '국어'
 - '이승철' '수학', '과학'
- 담당교과목명을 입력하면 교사명을 출력한다
- 담당교과목을 여러명의 교사가 강의하는 경우, 모든 교사명을 다 출력한다

연습문제 3 코드

```
def find_teacher(sub):
    Subjects = {'김경미':['수학','과학'],'최영희':['영어','수학'],'강동원':['영어','정필수':['사회','역사'],'박희수':['국어'],'이승철':['수학','과학']}

    teacher = []
    for key in Subjects.keys():
        if sub in Subjects[key]:
            teacher.append(key)

    return teacher

sub = input('과목을 입력하시오: ')
print(find_teacher(sub))
```

```
>>>
===== RESTART: E:/1_Works/2017Wc
과목을 입력하시오: 수학
['김경미', '최영희', '이승철']
>>>
===== RESTART: E:/1_Works/2017Wc
>>>
과목을 입력하시오: 영어
['최영희', '강동원']
>>> |
```

숙제

- 연습문제 2,3번을 입력한 코드와
- 실행 결과를 캡처하여 게시판에 올리시오

요약

- 딕셔너리에서 사용하는 메소드 활용한다
- 딕셔너리에서 여러 개의 값을 지정하고 활용한다
- 리스트로 구성된 딕셔너리를 활용한다

감사합니다

10주차_03 리스트튜플딕셔너리 활용