



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

□ 튜플

- 튜플은 값을 변경(추가, 삭제, 수정)시킬 수 없다는 점만 제외하면 리스트와 완전히 동일

```
t1 = (1, 2, 'a', 'b')
del t1[0]
```

TypeError: 'tuple' object doesn't support item deletion

```
t1 = (1, 2, 'a', 'b')
t1[1]=10
```

TypeError: 'tuple' object does not support item assignment

- 리스트는 [과]으로 둘러싸지만 튜플은 (과)으로 둘러싼다.
 - ▣ 리스트 : t1 = [1, 2, 'a', 'b']
 - ▣ 튜플 : t1 = (1, 2, 'a', 'b')



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 튜플 예제

□ 인덱싱

```
t1 = (1, 2, 'a', 'b')
i=t1[0]
j=t1[3]
print(i,j)
```

1 b

□ 슬라이싱

```
t1 = (1, 2, 'a', 'b')
i= t1[1:]
print(i)
```

(2, 'a', 'b')

□ 더하기

```
t1 = (1, 2, 'a', 'b')
t2 = (3, 4)
i = t1 + t2
print(i)
```

```
(1, 2, 'a', 'b', 3, 4)
```

□ 곱하기

```
t2 = (3, 4)
i = t2 * 3
print(i)
```

```
(3, 4, 3, 4, 3, 4)
```

□ 길이구하기

```
t1 = (1, 2, 'a', 'b')
i = len(t1)
print(i)
```

```
4
```



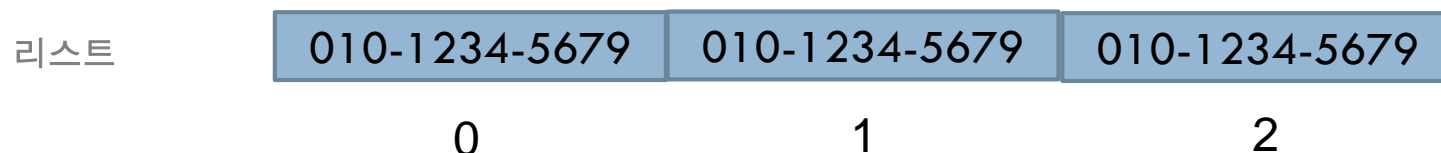
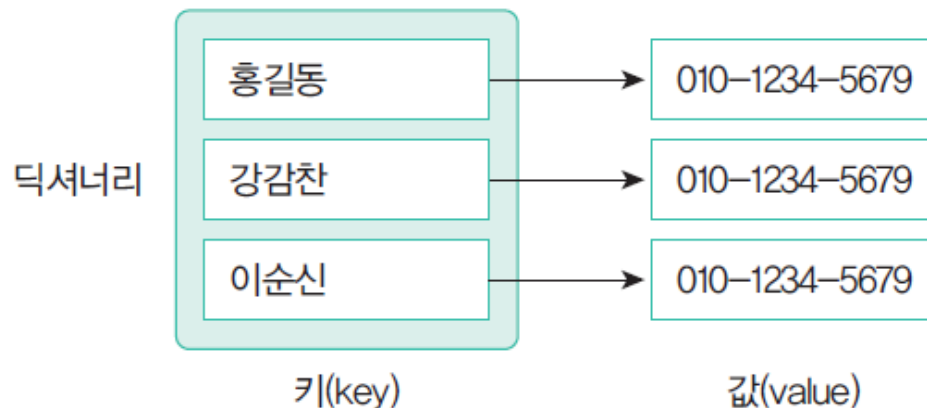
한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

□ 딕셔너리

- 딕셔너리(dictionary)도 리스트와 같이 값을 저장하는 방법이다. 하지만 딕셔너리에는 값(value)과 관련된 키(key)가 있다.
 - ▣ **key** 는 중복되어서는 안됨





한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - 데이터 추가

□ 추가하기

▣ 형식

```
변수명 = {key1:value1 , key2:value2 , . . . }
```

```
변수명 [key1]=value1
```

```
dic_val = {1: 'a', 3:'c'}  
print(dic_val)
```

```
dic_val[2] = 'b'  
print(dic_val)
```

```
{1: 'a', 3: 'c'}  
{1: 'a', 2:'b',3: 'c'}
```



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - 데이터 변경, 삭제, 값가져오기

□ 변경하기

▣ 형식

```
변수명[key1]=value1
```

```
dic_val = {1: 'a', 2: 'b', 3: 'c'}  
print(dic_val)
```

```
dic_val[2]='x'  
print(dic_val)
```

```
{1: 'a', 2: 'b', 3: 'c'}
```

```
{1: 'a', 2: 'x', 3: 'c'}
```

□ 삭제하기

■ 형식

```
del 변수명[key]
```

```
dic_val = {1: 'a', 2: 'b', 3: 'c'}  
print(dic_val)
```

```
del dic_val[1]  
print(dic_val)
```

```
{1: 'a', 2: 'b', 3: 'c'}  
{2: 'b', 3: 'c'}
```



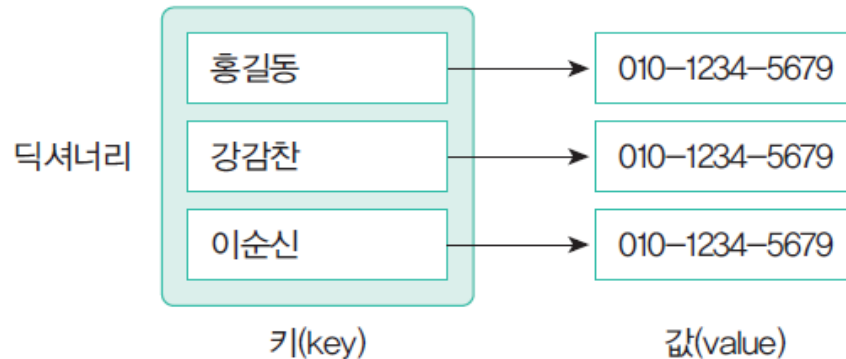
한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - 키를 이용 데이터 가져오기

- 딕셔너리에는 값(value)과 관련된 키(key)가 있다.
 - ▣ **key** 는 중복되어서는 안됨



- **value** 를 얻는 방법
 - ▣ 1. 키를 알고 있는 경우
 - ▣ 2. 키를 모르는 경우
 - 모든 **key**를 얻는 방법
 - **key**를 이용 **value**를 얻는다.
 - 모든 **value**를 얻는 방법

□ 딕셔너리에서 Key 사용해 Value 얻기

▣ 형식

저장변수 = 딕셔너리변수 [key]

```
grade = {'tom': 50, 'juli': 99}
```

```
gradeValue1 = grade['tom']
```

```
gradeValue2 = grade['juli']
```

```
print(gradeValue1, gradeValue2)
```

50 99



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

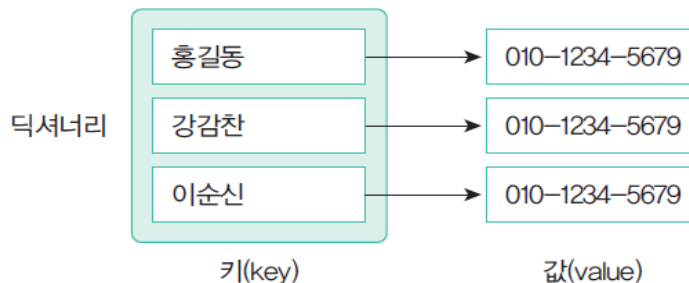
7장 딕셔너리와 튜플

- 딕셔너리 - 모든 키 가져오기

- **keys()** : 딕셔너리 Key만을 모아서 dict_keys라는 객체를 리턴
 - ▣ 리턴값 : dict_keys -리스트처럼 사용 가능
 - ▣ 형식 : 변수 = 딕셔너리변수.keys()

```
p_info = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}  
pKeys= p_info.keys()  
print(pKeys)
```

```
dict_keys(['name', 'phone', 'birth'])
```



■ dict_keys 객체를 리스트로 변환 방법

- 형식 : 변수 = list(dict_keys를 저장한 변수)

```
p_info = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}
pKeys = p_info.keys()
key_list=list(pKeys)
print(key_list)
for i in key_list:
    print(i)
```

```
['name', 'phone', 'birth']
name
phone
birth
```

- 리스트형태의 key들을 이용 value 얻기

```
변수 = list(dict_keys를 저장한 변수)
```

```
p_info = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}
pKeys = p_info.keys()
key_list=list(pKeys)
for i in key_list:
    p_val=p_info[i]
    print(p_val)
```

```
pey
0119993323
1118
```



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - 모든 value 가져오기

- `values ()` : 딕셔너리 `value`만을 모아서 `dict_values` 라는 객체를 리턴
 - `dict_values` 는 리스트처럼 사용 가능
 - 형식 : `변수 = 딕셔너리변수.values()`

```
score_dic = {'john': 100, 'mary': 90, 'danny': 80}  
score_values= score_dic.values()  
print(score_values)
```

```
dict_values([100, 90, 80])
```

| key | value |
|-------|-------|
| john | 100 |
| mary | 90 |
| danny | 80 |

- 딕셔너리 **value**들을 리스트로 변환
 - ▣ 리스트 관련 함수(sum, len, max, min) 모두 사용 가능

```
score_dic = {'john': 100, 'mary': 90, 'danny': 80}
score_dic_values= score_dic.values()
score_list=list(score_dic_values)
score_sum=sum(score_list)
score_len=len(score_list)
score_avg=score_sum/score_len
print(score_avg)
```

```
[100, 90, 80]
```



한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - clear 함수

print(b)

딕셔너리 관련 함수

24/36

- **clear ()** : 딕셔너리 안의 모든 요소를 삭제한다.
 - ▣ 형식 :

```
딕셔너리변수.clear()
```

```
a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}  
a.clear()  
print(a)
```

```
{}
```




한림대학교 SW중심대학

생초보를 위한 파이썬 프로그래밍

7장 딕셔너리와 튜플

- 딕셔너리 - in

- **in** : 해당 **key** 가 딕셔너리안에 있는지 조사함
 - ▣ 형식 :
`변수 = key in 딕셔너리저장변수`
 - ▣ 리턴값 :
 - **True** : 딕셔너리 안에 존재
 - **False** : 딕셔너리 안에 없음

```
a = {'name':'pey', 'phone':'0119993323', 'birth': '1118'}  
isNameExist= 'name' in a  
isEmailExist= 'email' in a  
print(isNameExist, isEmailExist)
```

True False

편의점 재고 관리 프로그램

27/36

- 편의점에서 재고 관리를 수행하는 프로그램을 작성해보자. 편의점에서 판매하는 물건의 재고를 딕셔너리에 저장한다.



| 물품명 | 재고량 |
|------|-----|
| 커피음료 | 7 |
| 펜 | 3 |
| 종이컵 | 2 |
| 우유 | 1 |
| 콜라 | 4 |
| 채 | 5 |

물건의 이름을 입력하시오: 콜라

4

Solution

```
items = { "커피음료": 7, "펜": 3, "종이컵": 2, "우유": 1, "콜라": 4, "책": 5 }
```

```
item = input("물건의 이름을 입력하시오: ");  
print (items[item])
```

- 영한사전을 구현해보자. 영어 단어를 키로 하고 설명을 값으로 하여 저장하면 될 것이다(one, two, three 를 한글로 출력)

단어를 입력하시오: one

하나

단어를 입력하시오: two

둘

Solution

30/36

```
english_dict = dict()

english_dict['one'] = '하나'
english_dict['two'] = '둘'
english_dict['three'] = '셋'

word = input("단어를 입력하시오: ");
print (english_dict[word])
```



도전문제

영한사전이 아닌 한영사전을 만들려면 어떻게 해야 하는가?

- 키보드로 **name, birth, age**값을 입력 받아 다음과 같은 딕셔너리를 만드세요

- ▣ **value**의 숫자는 모두 문자열로 저장

| 항목 | 값 |
|-------|------|
| name | 홍길동 |
| birth | 1128 |
| age | 30 |

- 홍길동이 태어난 달을 출력하세요
- **key**들중에 **email** 있는지 없는지 여부를 출력하세요
 - ▣ 있으면 : True
 - ▣ 없으면 : Fasle