

사전 연산

사전 연산

❖ 사전에 요소 추가하기

- 새로운 키를 사용하여 값을 배정

```
solar1 = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해  
왕성']  
solar2 = ['Sun', 'Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn',  
'Uranus', 'Neptune']  
  
solardict = {}  
for i, k in enumerate(solar1):  
    val = solar2[i]  
    solardict[k] = val  
  
print(solardict)
```

```
{'태양': 'Sun', '수성': 'Mercury', '금성': 'Venus', '지구': 'Earth',  
'화성': 'Mars', '목성': 'Jupiter', '토성': 'Saturn', '천왕성': 'Uranus',  
'해왕성': 'Neptune'}
```

사전 연산

❖ 사전의 특정 요소값 변경하기

- `dic[k]=val2`

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}  
names['Aimy'] = 10000  
print(names)
```

```
{'Mary': 10999, 'Sams': 2111, 'Aimy': 10000, 'Tom': 20245, 'Michale': 27115,  
'Bob': 5887, 'Kelly': 7855}
```

사전 연산

❖ 사전의 특정 요소 제거하기(del)

- o `del dic[k]`

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}  
del names['Sams']  
print(names)
```

```
{'Mary': 10999, 'Aimy': 9778, 'Tom': 20245, 'Michale': 27115, 'Bob': 5887,  
'Kelly': 7855}
```

사전 연산

❖ 사전의 모든 요소 제거하기(clear)

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}
```

```
names.clear()
```

```
print(names)
```

```
{}
```

사전 연산

❖ 사전에서 키만 추출하기(keys)

- keys 메소드 : 모든 키를 추출하여 dict_keys 사전 뷰 객체로 리턴
 - dict_keys 객체는 list()로 리스트 형변환 가능

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
'Michale':27115, 'Bob':5887, 'Kelly':7855}
```

```
ks = names.keys()
```

```
print(ks)
```

```
for k in ks:
```

```
    print('Key:%s \tValue:%d' %(k, names[k]))
```

```
dict_keys(['Mary', 'Sams', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly'])
```

```
Key:Mary   Value:10999
```

```
Key:Sams   Value:2111
```

```
Key:Aimy   Value:9778
```

```
Key:Tom    Value:20245
```

```
Key:Michale Value:27115
```

```
Key:Bob    Value:5887
```

```
Key:Kelly   Value:7855
```

사전 연산

❖ 사전에서 값만 추출하기(values)

- 사전에서 모든 값을 추출하여 dict_values 사전 뷰 객체로 리턴

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,
'Michale':27115, 'Bob':5887, 'Kelly':7855}
vals = names.values()
print(vals)

vals_list = list(vals)
ret = sum(vals_list)
print('출생아수 총계: %d' %ret)
```

```
dict_values([10999, 2111, 9778, 20245, 27115, 5887, 7855])
출생아수 총계: 83990
```

사전 연산

❖ 사전 요소를 모두 추출하기(items)

- 사전의 요소를 모두 추출하여 (키, 값)의 쌍으로 구성되는 dict_items 사전 뷰 객체를 리턴

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}  
items = names.items()  
print(items)  
  
for item in items:  
    print(item)
```

```
dict_items([('Mary', 10999), ('Sams', 2111), ('Aimy', 9778), ('Tom', 20245),  
('Michale', 27115), ('Bob', 5887), ('Kelly', 7855)])  
( 'Mary', 10999)  
( 'Sams', 2111)  
( 'Aimy', 9778)  
( 'Tom', 20245)  
( 'Michale', 27115)  
( 'Bob', 5887)  
( 'Kelly', 7855)
```


사전 연산

❖ 사전에 특정 키가 존재하는지 확인하기(in)

- o `k in dict` → True/False 리턴

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
'Michale':27115, 'Bob':5887, 'Kelly':7855}  
k = input('이름을 입력하세요: ')  
if k in names:  
    print('이름이 <%s>인 출생아수는 <%d>명 입니다.' %(k, names[k]))  
else:  
    print('자료에 <%s>인 이름이 존재하지 않습니다.' %k)
```

사전 연산

❖ 사전에 특정 값이 존재하는지 확인하기(in)

- dict.values() 메서드 활용

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
'Michale':27115, 'Bob':5887, 'Kelly':7855}
```

```
if 2111 in names.values() :  
    print('이름이 <%s>인 출생아수는 <%d>명 입니다.' %(k, names[k]))  
else:  
    print('자료에 <%s>인 이름이 존재하지 않습니다.' %k)
```

사전 연산

❖ 사전 정렬하기(sorted)

- sorted(dict) 함수
 - 기본적으로 사전의 키를 오름차순으로 정렬
 - 정렬된 값의 리스트를 리턴

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}  
ret1 = sorted(names)  
print(ret1)
```

```
['Aimy', 'Bob', 'Kelly', 'Mary', 'Michale', 'Sams', 'Tom']
```

사전 연산

❖ 사전 정렬하기(sorted)

- sorted() 함수에 key 인자(정렬 기준 함수 참조)를 추가하여 정렬 기준 변경 가능

```
names = {  
    'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,  
    'Michale':27115, 'Bob':5887, 'Kelly':7855  
}  
  
def f1(x):  
    return x[0]        // 첫 번째 요소로 비교 - 이 경우 키  
  
def f2(x):  
    return x[1]        // 두 번째 요소로 비교 - 이 경우 값  
  
ret2 = sorted(names.items(), key=f1)  
print(ret2)  
  
ret3 = sorted(names.items(), key=f2)  
print(ret3)  
  
ret4 = sorted(names.items(), key=f2, reverse=True)  
print(ret4)
```