



딕셔너리 자료형

Index

[딕셔너리란?](#)

[딕셔너리는 어떻게 만들까?](#)

[딕셔너리 쌍 추가, 삭제하기](#)

[딕셔너리 쌍 추가하기](#)

[딕셔너리 요소 삭제하기](#)

[딕셔너리를 사용하는 방법](#)

[딕셔너리에서 Key를 사용해 Value 얻기](#)

[딕셔너리 만들 때 주의할 사항](#)

[딕셔너리 관련 함수](#)

[Key 리스트 만들기 - keys](#)

[Value 리스트 만들기 - values](#)

[Key, Value 쌍 얻기 - items](#)

[Key: Value 쌍 모두 지우기 - clear](#)

[Key로 Value 얻기 - get](#)

[해당 Key가 딕셔너리 안에 있는지 조사하기 - in](#)

딕셔너리란?

- 대응 관계를 나타내는 자료형 , 사전, Key를 통해 Value를 얻는다.

딕셔너리는 어떻게 만들까?

```
{Key1: Value1, Key2: Value2, Key3: Value3, ...}
```

```
dic = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}
```

key	value
name	pey
phone	010-9999-1234
birth	1118

```
a = {1: 'hi'}
```

```
a = {'a' : [1, 2, 3]}
```

딕셔너리 쌍 추가, 삭제하기

딕셔너리 쌍 추가하기

```
a = {1 : 'a'}
a[2] = 'b'
a
>>> {1: 'a', 2: 'b'}
```

```
a['name'] = 'pey'
a
>>> {1: 'a', 2: 'b', 'name': 'pey'}
```

```
a[3] = [1, 2, 3]
a
>>> {1: 'a', 2: 'b', 'name': 'pey', 3: [1, 2, 3]}
```

딕셔너리 요소 삭제하기

```
del a[1]
a
>>> {2: 'b', 'name': 'pey', 3: [1, 2, 3]}
```

딕셔너리를 사용하는 방법

딕셔너리에서 Key를 사용해 Value 얻기

```
grade = {'pey': 10, 'julliet': 99}
grade['pey']
>>> 10
```

```
grade['julliet']  
>>> 99
```

```
a = {1:'a', 2:'b'}  
a[1]  
>>> 'a'  
a[2]  
>>> 'b'
```

```
a = {'a': 1, 'b': 2}  
a['a']  
>>> 1  
a['b']  
>>> 2
```

```
dic = {'name':'pey', 'phone':'010-9999-1234', 'birth': '1118'}  
dic['name']  
>>> 'pey'  
dic['phone']  
>>> '010-9999-1234'  
dic['birth']  
>>> '1118'
```

딕셔너리 만들 때 주의할 사항

- 중복되는 Key 값을 설정해 놓으면 하나를 제외한 나머지 것들이 모두 무시한다.
- Key에 리스트는 쓸 수 없다.

```
a = {1:'a',1:'b'}  
a = {[1,2] : 'hi'}
```

딕셔너리 관련 함수

Key 리스트 만들기 - keys

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}  
a.keys()  
>>> dict_keys(['name', 'phone', 'birth'])
```

```
list(a.keys())
```

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

Value 리스트 만들기 - values

```
a.values()  
>>> dict_values(['pey', '010-9999-1234', '1118'])
```

Key, Value 쌍 얻기 - items

```
a.items()  
>>> dict_items([('name', 'pey'), ('phone', '010-9999-1234'), ('birth', '1118')])
```

Key: Value 쌍 모두 지우기 - clear

```
a.clear()  
a  
>>> {}
```

Key로 Value 얻기 - get

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}  
a.get('name')  
>>> 'pey'  
a.get('phone')  
>>> '010-9999-1234'
```

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}  
print(a.get('nokey'))  
>>> None  
print(a['nokey'])  
>>> KeyError: 'nokey'
```

```
a.get('nokey', 'foo')  
>>> 'foo'
```

해당 Key가 딕셔너리 안에 있는지 조사하기 - in

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}  
'name' in a  
>>> True  
'email' in a  
>>> False
```