

5. 딕셔너리 자료형

5.1. 딕셔너리란

- 이름 = 오나희, 생일 = 8월 24일
 - 대응 관계를 나타낼 수 있는 자료형
- 리스트나 튜플처럼 순차적으로 해당 요솟값을 구하지 않고 key를 통해 value를 얻는다
 - baseball = 야구를 찾는다면 사전의 내용을 모두 순차적으로 검색하는 것이 아니라 이 단어가 있는 곳만 보는 것

5.2. 딕셔너리는 어떻게 만들까

딕셔너리 기본 모습

```
hee = {'name' : 'hee', 'phone': 'iphone', 'birth':'0824'}
```

key	value
name	hee
phone	iphone
birth	0824

딕셔너리 쌍 추가, 삭제하기

```
hee = {'name' : 'hee', 'phone': 'iphone', 'birth':'0824'}  
hee['emotion'] = 'happy'  
print(hee)
```

- key 가 emtion , value 가 happy인 쌍이 추가되었다

```
hee = {'name' : 'hee', 'phone': 'iphone', 'birth':'0824'} del hee['name'] print(hee)
```

- del [키] 로 삭제할 수 있다

5.3. 딕셔너리를 사용하는 방법

- 예를 들어 4명의 사람이 있다고 가정하고 각자의 특기를 표현할 수 있는 좋은 방법에 대해 생각해보면 리스트나 문자열로 표현하기 까다롭다

```
{"오지훈": "키보드수집가", "오나희": "서트부장", "이호섭": "호가오리"}
```

하지만 이런 식으로 사람 이름과 특기를 한 쌍으로 묶은 딕셔너리는 간편하다

딕셔너리에서 key 를 사용해 value 얻기\

```
grade = {'pay' : 10}  
print(grade['pay'])
```

- 10이 출력된다
- pay 라는 키의 value 를 얻기 위해 grade[pay] 를 사용한 것 처럼 어떤 키의 값을 얻기 위해서는 딕셔너리 변수 이름[key] 를 사용해야 한다

딕셔너리를 만들 때 주의할 사항

- 딕셔너리에서 키는 고유한 값이므로 키 값을 설정해 놓으면 하나를 제외한 나머지 것들이 모두 무시된다

```
a = {1:'a', a:'b'}  
print(a)
```

출력 {1 'b'}

5.4. 딕셔너리 관련 함수

Key 리스트 만들기

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

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

- 키 값만을 리턴한다

Value 리스트 만들기

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

출력 dict_values(['pey', '010-9999-1234', '1118'])

- 아까와는 반대로 value 값만을 리턴한다

Key, Value 쌍 얻기

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

출력 : dict_items([('name', 'pey'), ('phone', '010-9999-1234'), ('birth', '1118')])

- items 함수는 Key 와 Value 의 쌍을 튜플로 묶은 값을 dict_items 객체로 리턴한다

Key : Value 쌍 모두 지우기

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

출력 : {}

- 딕셔너리 안의 모든 요소를 삭제한다

Key 로 Value 얻기

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

출력 : 'pey'

- get(x) 라는 함수는 x라는 key에 대응되는 value를 리턴한다

해당 Key 가 딕셔너리 안에 있는지 조사하기

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}  
  
print('name' in a)
```

출력 : True

- `name`의 문자열은 `a` 딕셔너리의 `key` 중 하나이다 따라서 `name in a` 를 호출하면 `Ture (참)` 을 리턴한다