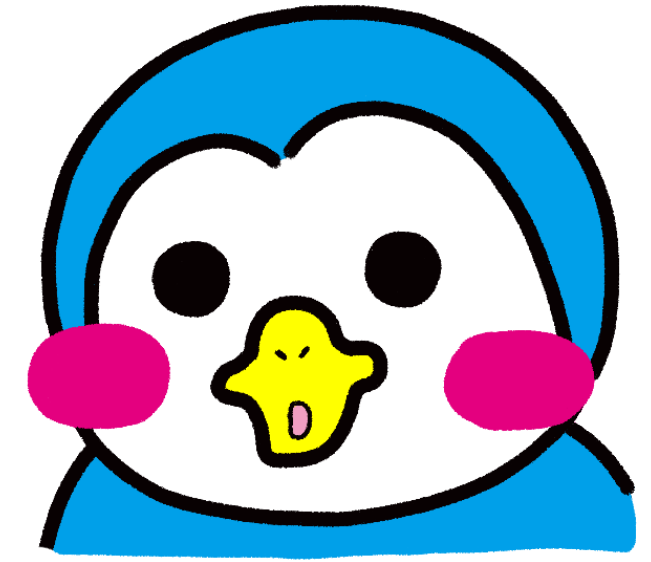


입문자를 위한 파이썬 기초



딕셔너리(dict) 알아보기

자료형 딕셔너리를 소개합니다

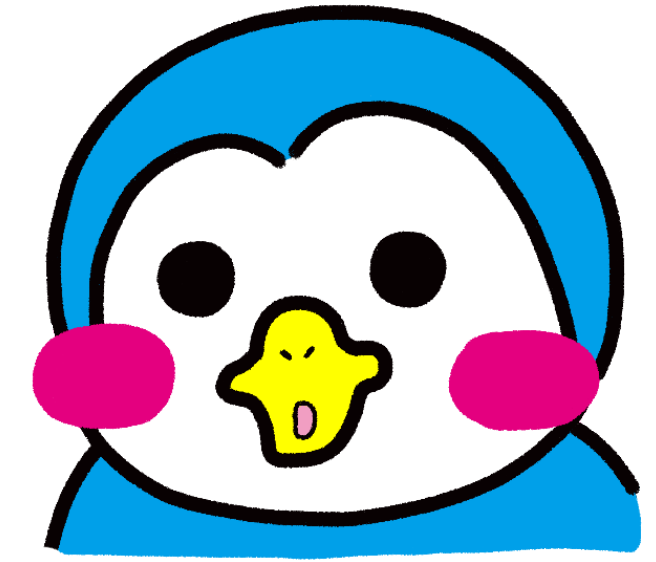


문자열, 리스트, 튜플 타입의 공통점은 ‘여러 개의 멤버(값)를 가진다’는 점이다. 그리고 파이썬의 또 다른 기본 자료형인 딕셔너리(dictionary) 또한 이들과 마찬가지로 여러 개의 멤버를 가질 수 있는 타입이다.

딕셔너리는 멤버를 중괄호로 감싸는 형태로 만들어진다.

```
tu = ("apple", "banana") # 튜플은 소괄호  
li = ["apple", "banana"] # 리스트 대괄호  
di = {} # 딕셔너리는 중괄호를 사용합니다.
```

딕셔너리의 멤버

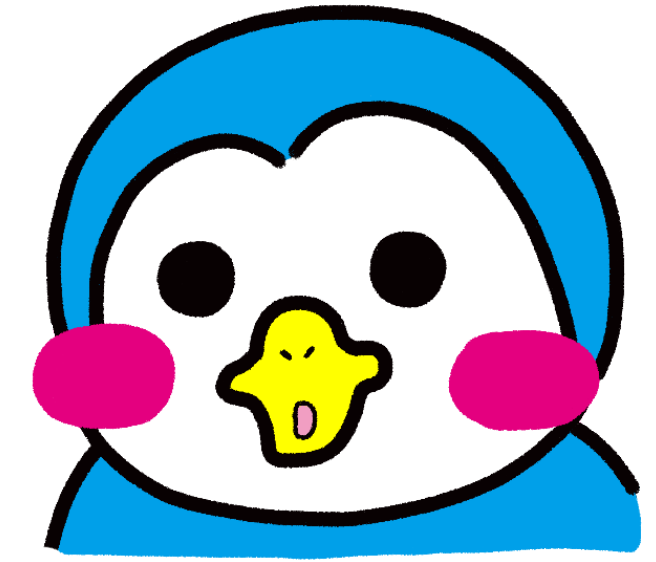


딕셔너리의 멤버들은 각자가 두 개의 값을 가진다! 이 중 하나를 키(key), 또 다른 하나를 밸류(value)라 표현한다. 키는 멤버의 이름을 나타내는 역할을, 밸류는 멤버가 가진 실제 값을 나타내는 역할을 한다.

```
di = {  
    ...  
    key: value  
}
```

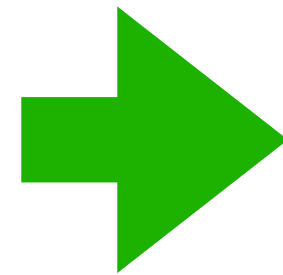
=> 위 딕셔너리는 멤버가 하나 있는 상태!

딕셔너리의 멤버



딕셔너리의 멤버들은 각자가 두 개의 값을 가진다! 이 중 하나를 키(key), 또 다른 하나를 밸류(value)라 표현한다. 키는 멤버의 이름을 나타내는 역할을, 밸류는 멤버가 가진 실제 값을 나타내는 역할을 한다.

```
di = {  
    ...  
    key: value  
}
```

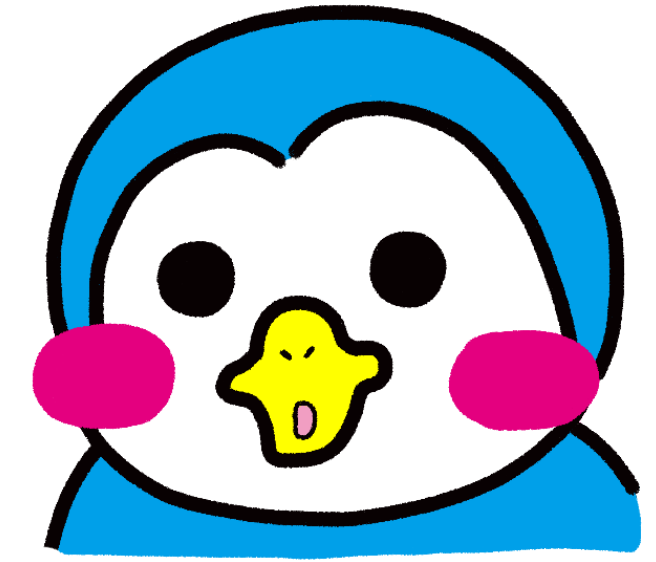


```
di = {  
    ...  
    key1: value1 ,  
    key2: value2  
}
```

나는야 멤버가
두 개인 딕셔너리!

=> 멤버 간의 구분을 위해서는 쉼표(,)를 이용하면 된다!

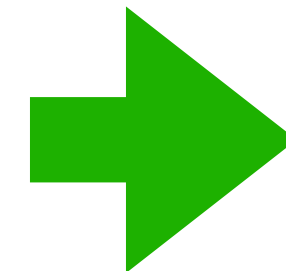
키를 왜 붙이는 건데!?



딕셔너리는 여러 개의 멤버(데이터)를 가질 수 있지만, 인덱스 번호를 가지고 있지 않다. 대신 키가 그 역할을 수행할 수 있다. 멤버마다 번호가 아닌 이름(키)을 가지고 있는 형태이기 때문에, 보다 직관적으로 데이터를 관리할 수 있다.

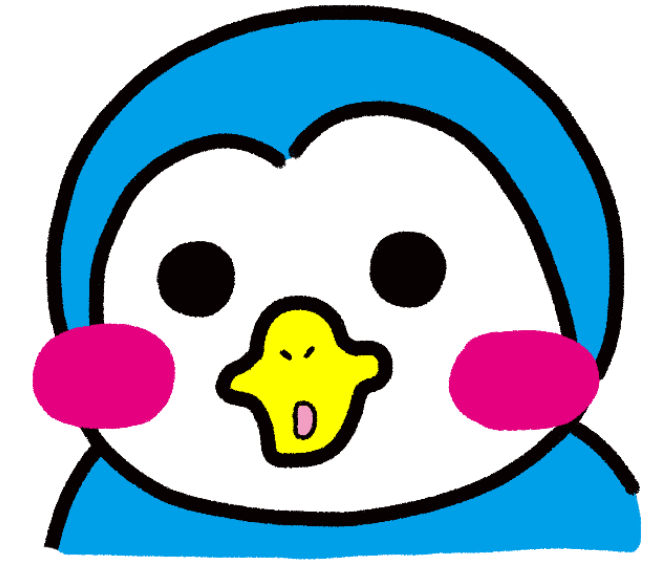
```
boy = {  
    "name": "유노코딩",  
    "residence": "대전광역시",  
    "job": "IT 프리랜서"  
}
```

```
print(boy["residence"])
```



대전광역시

키를 왜 붙이는 건데!?

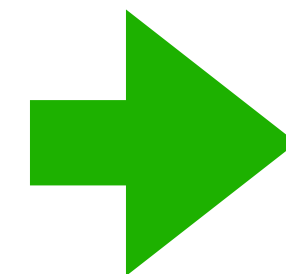


딕셔너리는 여러 개의 멤버(데이터)를 가질 수 있지만, 인덱스 번호를 가지고 있지 않다. 대신 키가 그 역할을 수행할 수 있다. 멤버마다 번호가 아닌 이름(키)을 가지고 있는 형태이기 때문에, 보다 직관적으로 데이터를 관리할 수 있다.

```
boy = {  
    "name": "유노코딩",  
    "residence": "대전광역시",  
    "job": "IT 프리랜서"  
}
```

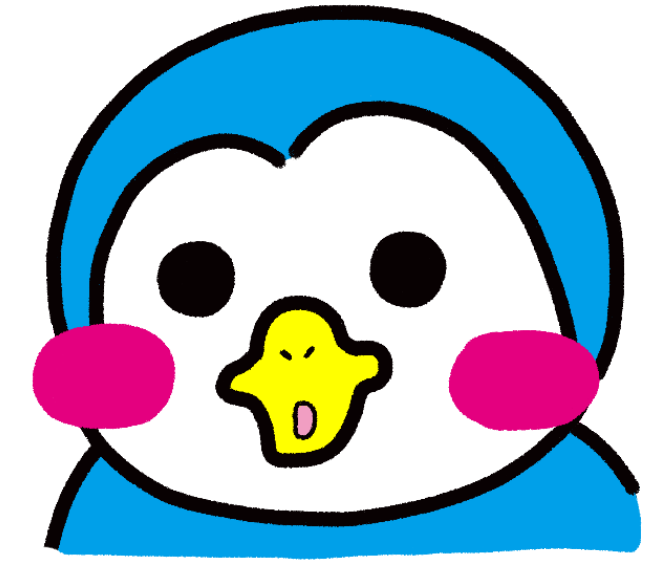
```
print(boy["residence"])
```

키는 각 멤버에 접근할 수 있는, 멤버의 대표 값이다.
따라서 딕셔너리 멤버들 간의 키 중복은 허용되지 않는다.
키가 같은 멤버가 있을 경우, 그 중 하나의 멤버만 살아남는다.
단, 멤버들 간에 밸류가 같은 것은 문제가 되지 않는다.



대전광역시

키와 밸류

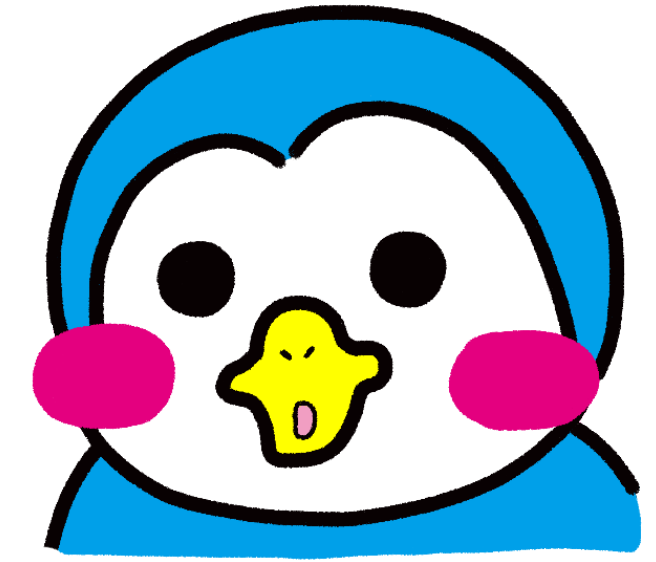


딕셔너리의 멤버는 키로 변경 불가능한 자료형, 즉 immutable 자료형만 사용할 수 있다. 반면 밸류로는 어떤 자료형이라도 사용할 수 있다.

```
korea2010 = {  
    7: "박지성",  
    "감독": "허정무"  
}
```

```
korea2022 = {  
    [7, 11] : ("손흥민", "황희찬")  
}
```

키와 밸류

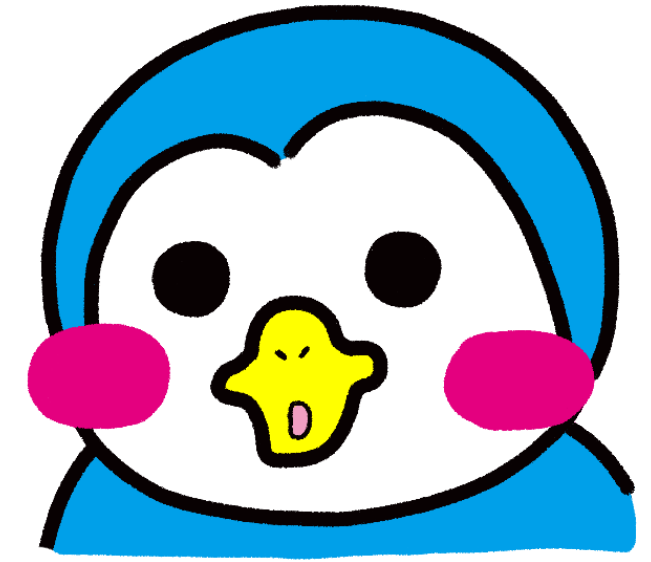


딕셔너리의 멤버는 키로 변경 불가능한 자료형, 즉 immutable 자료형만 사용할 수 있다. 반면 밸류로는 어떤 자료형이라도 사용할 수 있다.

```
korea2010 = {  
    "감독": "허정모",  
    "배우": "박지성",  
}
```

```
korea2022 = {  
    [7, 11]: "홍민", "황희찬"  
}
```


키와 밸류



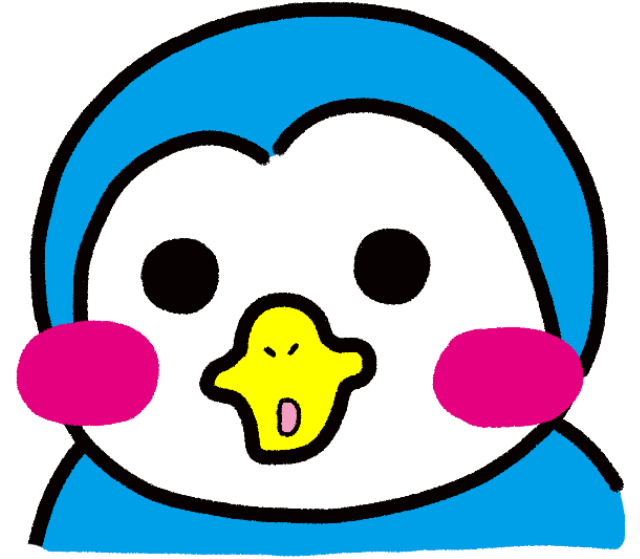
딕셔너리의 멤버는 키로 변경 불가능한 자료형, 즉 immutable 자료형만 사용할 수 있다. 반면 밸류로는 어떤 자료형이라도 사용할 수 있다.

```
korea2010 = {  
    "감독": "허정모",  
    "감독": "박지성",  
}
```

```
korea2022 = {  
    [7, 11]: "홍민", "황희찬"  
}
```

=> 그렇다면 딕셔너리는 변경 가능할까, 불가능할까? 답은.. 가능하다!

수정이나 삭제할 때도 키를 써요



```
boy = {  
    "name": "유노코딩",  
    "residence": "대전광역시",  
    "job": "IT 프리랜서"  
}
```

`boy["pet"] = "고양이"`

존재하지 않는 키를 입력 후
객체를 대입하면, 멤버가 추가된다!

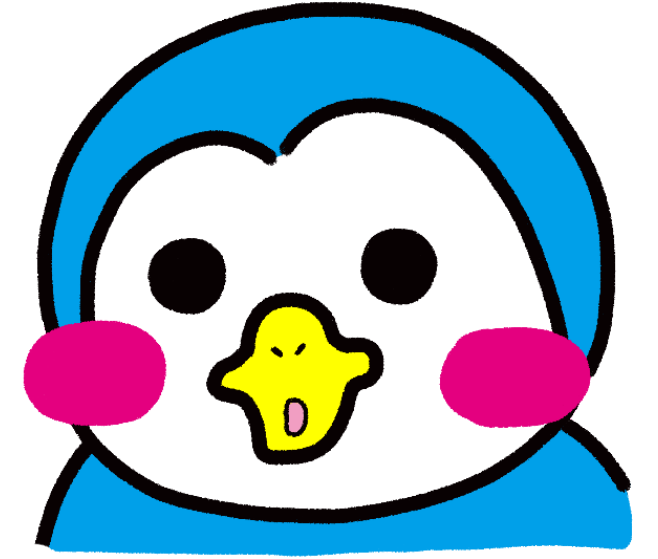
`del boy["name"]`

del 명령을 이용하면
특정 키를 가진 멤버를 삭제할 수 있다!

`boy["job"] = "크리에이터"`

이미 존재하는 키를 입력 후
객체를 대입하면, 해당 키의 밸류가 변경된다!

내용 정리



- 딕셔너리는 다수의 멤버를 가질 수 있는 타입으로, 중괄호로 멤버를 감싼다.
- 딕셔너리의 각 멤버는 키와 밸류, 두 개의 값을 가진다.
- 딕셔너리는 여러 개의 멤버를 가질 수 있지만, 인덱스 번호를 가지고 있지 않다. 대신 키가 그 역할을 수행할 수 있다.
- 키로는 변경 불가능한 자료형만 사용할 수 있다. 반면 밸류로는 어떤 자료형이라도 사용할 수 있다.
- 딕셔너리는 변경 가능한 자료형, 즉 mutable 자료형이다.