

점프 투 파이썬

02장 파이썬 프로그래밍의 기초, 자료형

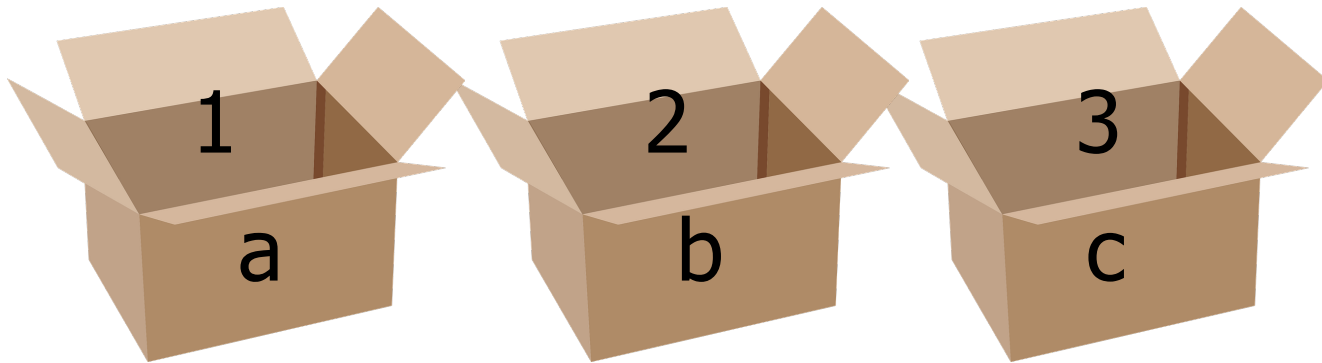
지은이: 박응용

강의: 조코딩

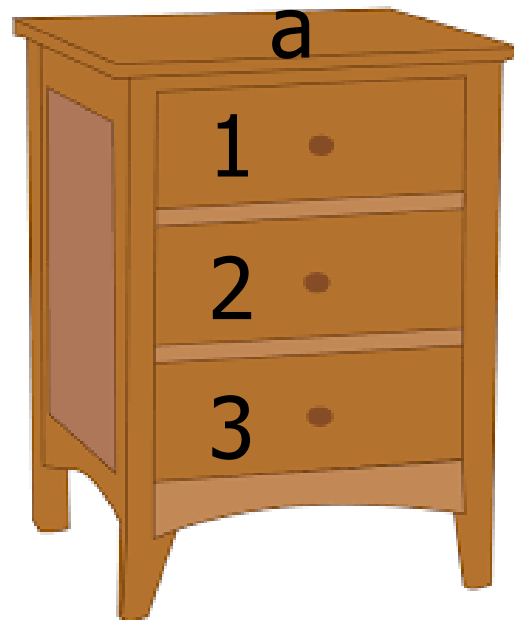
숫자, 문자열, 불

변수, 리스트, 튜플, 딕셔너리, 집합

리스트를 쓰는 이유?



리스트



튜플

리스트



튜플



02-4 튜플 자료형

튜플 요소값 삭제 시 오류

```
>>> t1 = (1, 2, 'a', 'b')
>>> del t1[0]
Traceback (innermost last):
  File "", line 1, in ?del t1[0]
TypeError: object doesn't support item deletion
```

02-4 튜플 자료형

튜플 요소값 변경 시 오류

```
>>> t1 = (1, 2, 'a', 'b')
>>> t1[0] = 'c'
Traceback (innermost last):
File "", line 1, in ?t1[0] = 'c'
TypeError: object doesn't support item assignment
```


02-4 튜플 자료형

인덱싱

```
>>> t1 = (1, 2, 'a', 'b')
>>> t1[0]
1
>>> t1[3]
'b'
```

슬라이싱

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

02-4 튜플 자료형

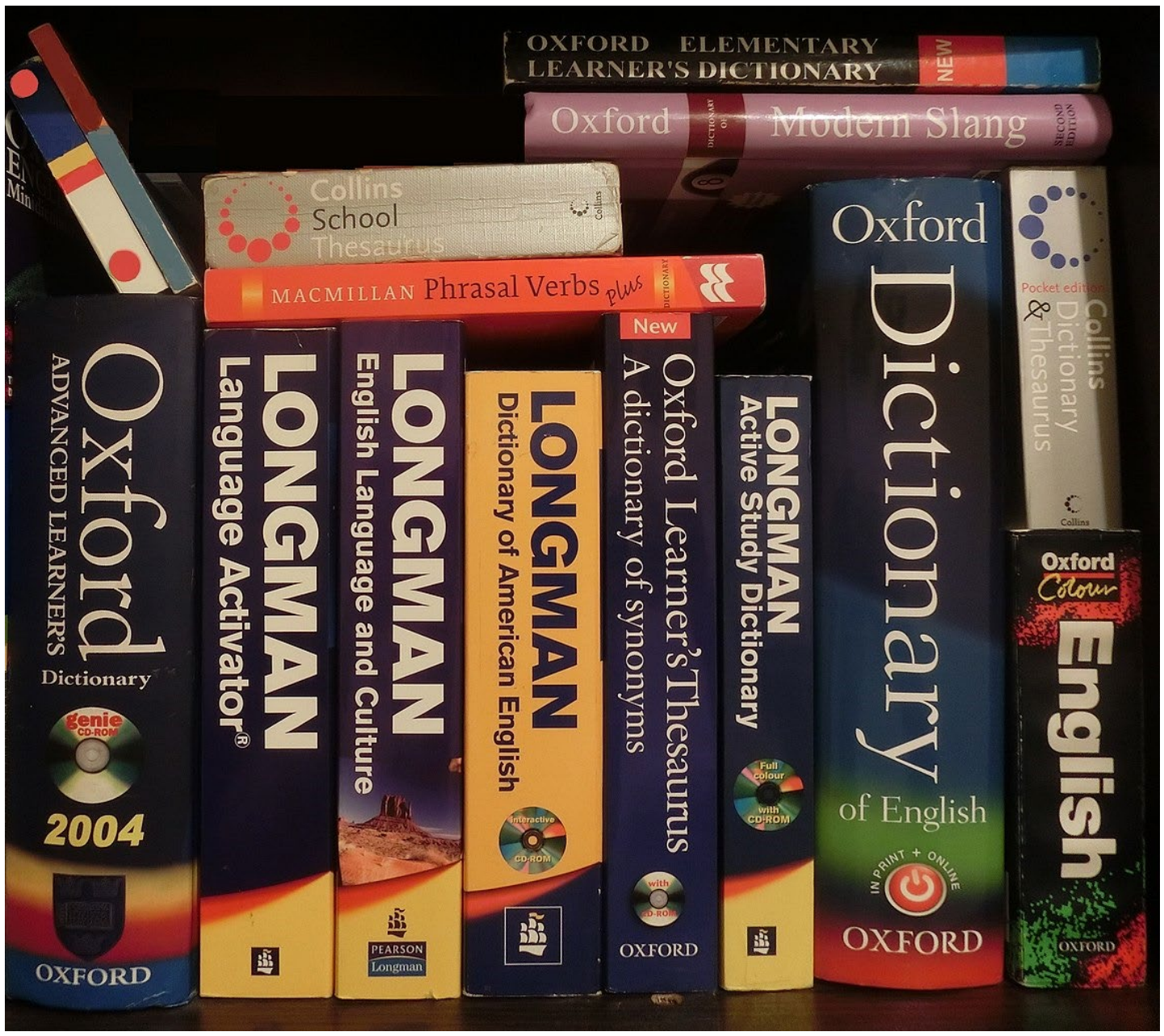
더하기

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

곱하기

```
>>> t2 * 3
(3, 4, 3, 4, 3, 4)
```

딕셔너리



with faces
games or
) one of
e played
at risks,
cut into

risky,

es) di-
efined
ed to

[Latin: related to *dictate*]
dictatorial /ˈdɪktəˈtɔːriəl/ *adv.*
like a dictator. 2 overbearing.
dictatorially *adv.* [Latin: related
TATOR]

diction /ˈdɪkʃ(ə)n/ *n.* manner
ciation in speaking or singing
dictio from *dico dict-* say]

dictionary /ˈdɪkʃənəri/ *n.* (pl)
book listing (usu. alphabetical)
explaining the words of a lan
giving corresponding words in
language. 2 reference book e
the terms of a particular

어학사전

dictionary



영어사전 단어·숙어 1-5 / 526건

dictionary 미국식 [-neri] 영국식 ['dikʃənri] ★★

1. (일반적인) 사전
2. (특정 분야의) 사전
3. (컴퓨터의) 사전

Hash

Map

Object

JSON

딕셔너리

a =



API에 자주 활용됨

예제 [편집]

다음은 한 사람에 관한 정보를 갖는 JSON 객체이다.

키-값 쌍(이름:값)의 패턴으로 표현된다.

```
1 {  
2   "이름": "홍길동",  
3   "나이": 25,  
4   "성별": "여",  
5   "주소": "서울특별시 양천구 목동",  
6   "특기": ["농구", "도술"],  
7   "가족관계": {"#": 2, "아버지": "홍판서", "어머니": "춘섬"},  
8   "회사": "경기 수원시 팔달구 우만동"  
9 }
```


02-5 딕셔너리 자료형

- 연관 배열(Associative array) 또는 해시(Hash)
- 단어 그대로 해석하면 사전이라는 뜻
- Key를 통해 Value를 얻는다

```
>>> dic = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}
```

02-5 딕셔너리 자료형

딕셔너리 쌍 추가하기

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

딕셔너리 요소 삭제하기

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

02-5 딕셔너리 자료형

딕셔너리에서 Key 사용해 Value 얻기

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

딕셔너리 만들 때 주의할 사항

```
>>> a = {1:'a', 1:'b'}
>>> a
{1: 'b'}
```

02-5 딕셔너리 자료형

Key 리스트 만들기(keys)

```
>>> a = {'name': 'pey', 'phone': '0119993323', 'birth':  
        '1118'}  
>>> a.keys()  
dict_keys(['name', 'phone', 'birth'])
```

Value 리스트 만들기(values)

```
>>> a.values()  
dict_values(['pey', '0119993323', '1118'])
```

02-5 딕셔너리 자료형

Key, Value 쌍 얻기(items)

```
>>> a = {'name': 'pey', 'phone': '0119993323', 'birth':  
'1118'}  
>>> a.items()  
dict_items([('name', 'pey'), ('phone', '0119993323'),  
('birth', '1118')])
```

Key: Value 쌍 모두 지우기(clear)

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

02-5 딕셔너리 자료형

Key로 Value얻기(get)

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

02-5 딕셔너리 자료형

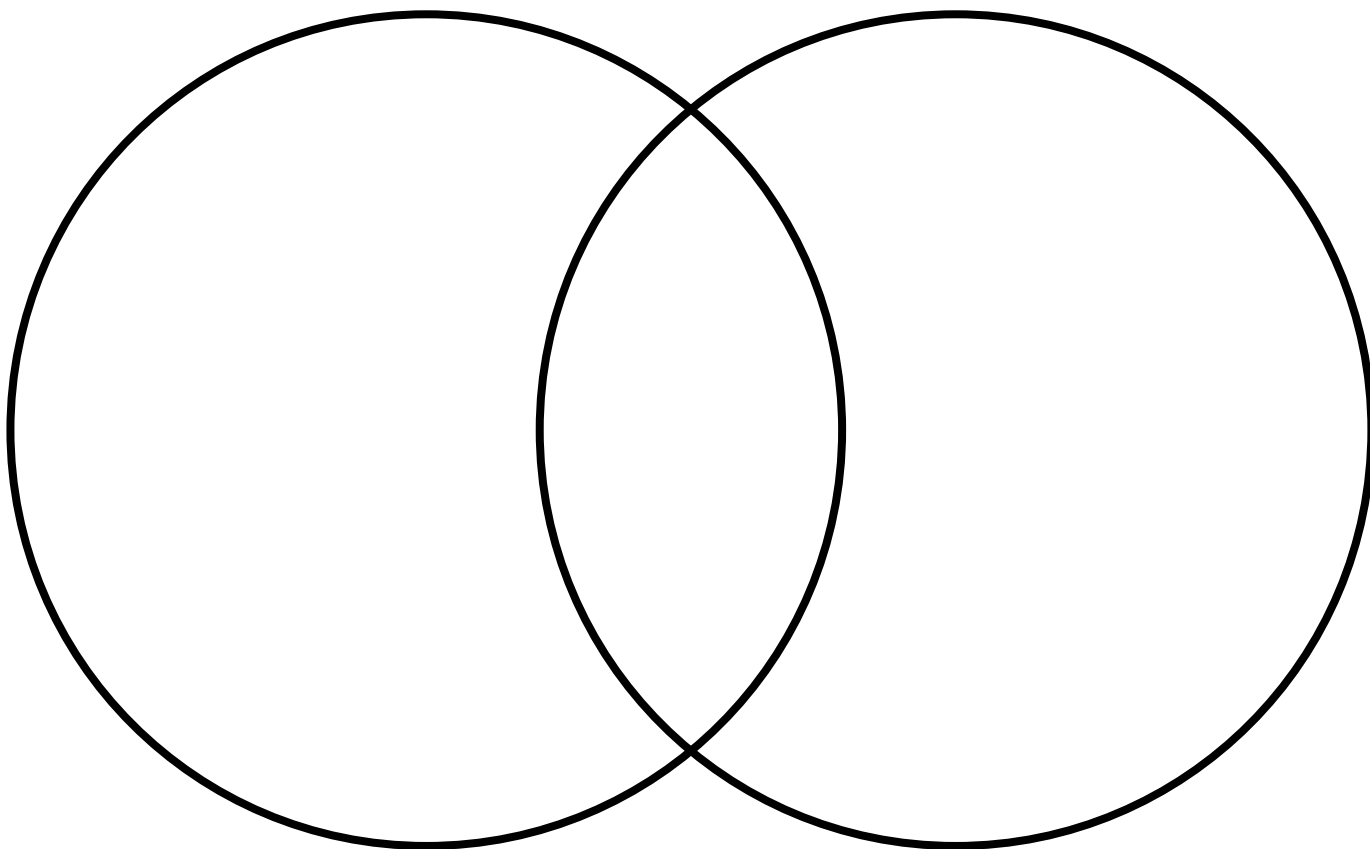
해당 Key가 딕셔너리 안에 있는지 조사하기(in)

```
>>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}  
>>> 'name' in a  
True  
>>> 'email' in a  
False
```

집합

A

B



02-6 집합 자료형

- 집합에 관련된 것들을 쉽게 처리하기 위해 만들어진 자료형
- 중복을 허용하지 않는다.
- 순서가 없다(Unordered).

02-6 집합 자료형

집합 자료형

```
>>> s1 = set([1,2,3])  
>>> s1  
{1, 2, 3}
```

순서가 없고 중복이 허용되지 않는다

```
>>> s2 = set("Hello")  
>>> s2  
{'e', 'l', 'o', 'H'}
```

02-6 집합 자료형

교집합 1

```
>>> s1 = set([1, 2, 3, 4, 5, 6])  
>>> s2 = set([4, 5, 6, 7, 8, 9])  
>>> s1 & s2  
{4, 5, 6}
```

교집합 2

```
>>> s1.intersection(s2)  
{4, 5, 6}
```

02-6 집합 자료형

합집합 1

```
>>> s1 = set([1, 2, 3, 4, 5, 6])  
>>> s2 = set([4, 5, 6, 7, 8, 9])  
>>> s1 | s2  
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

합집합 2

```
>>> s1.union(s2)  
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

02-6 집합 자료형

차집합 1

```
>>> s1 = set([1, 2, 3, 4, 5, 6])
>>> s2 = set([4, 5, 6, 7, 8, 9])
>>> s1 - s2
{1, 2, 3}
>>> s2 - s1
{8, 9, 7}
```

차집합 2

```
>>> s1.difference(s2)
{1, 2, 3}
>>> s2.difference(s1)
{8, 9, 7}
```

02-6 집합 자료형

값 1개 추가하기(add)

```
>>> s1 = set([1, 2, 3])
>>> s1.add(4)
>>> s1
{1, 2, 3, 4}
```

값 여러 개 추가하기(update)

```
>>> s1 = set([1, 2, 3])
>>> s1.update([4, 5, 6])
>>> s1
{1, 2, 3, 4, 5, 6}
```

02-6 집합 자료형

특정 값 제거하기(remove)

```
>>> s1 = set([1, 2, 3])  
>>> s1.remove(2)  
>>> s1  
{1, 3}
```


불(bool)

참(True)

거짓(False)

```
if a == 3:
```

```
    print(a)
```

```
while a == 3:
```

```
    a += 1
```

02-7 자료형의 참과 거짓

값	참 or 거짓
"python"	참
""	거짓
[1, 2, 3]	참
[]	거짓
()	거짓
{}	거짓
1	참
0	거짓
None	거짓

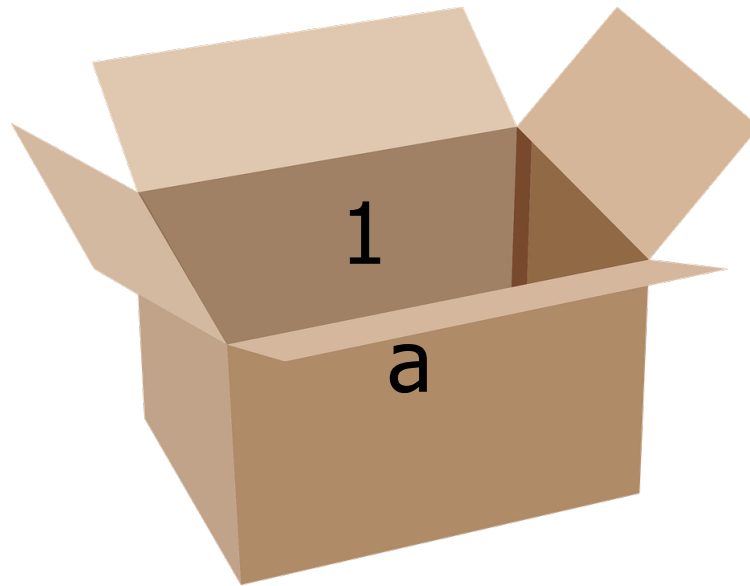
02-7 자료형의 참과 거짓

자료형의 참과 거짓은 어떻게 사용되나?

```
>>> a = [1, 2, 3, 4]
>>> while a:
...     a.pop()
...
4
3
2
1
```

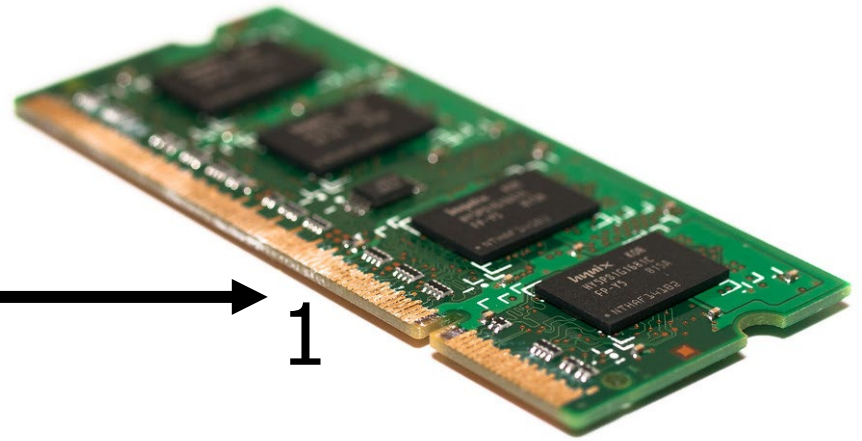
변수

변수



변수

메모리



주소: 2026043360
값: 1 (객체)

02-8 자료형의 값을 저장하는 공간, 변수

다음 예와 같은 a, b, c를 변수라고 한다.

```
>>> a = 1  
>>> b = "python"  
>>> c = [1,2,3]
```

변수를 만들 때는 =(assignment) 기호를 사용한다.

02-8 자료형의 값을 저장하는 공간, 변수

파이썬에서 사용하는 변수는 객체를 가리키는 것

```
>>> a = 3
```

- 3이라는 값을 가지는 정수 자료형(객체)이 자동으로 메모리에 생성
- 변수 a는 객체가 저장된 메모리의 위치를 가리키는 레퍼런스(Reference)
- a라는 변수는 3이라는 정수형 객체를 가리키고 있다

02-8 자료형의 값을 저장하는 공간, 변수

리스트 변수 주의사항

```
>>> a = [1,2,3]
>>> b = a
>>> a[1] = 4
>>> a
[1, 4, 3]
>>> b
[1, 4, 3]
```