Kümeler

Prof.Dr. Bahadır AKTUĞ
JFM212 Python ile Mühendislik Uygulamaları

*Kaynakça bölümünde verilen kaynaklardan derlenmiştir.

Kümeler (Sets)

- Kümeler, listelere benzer şekilde farklı tipte (metin tipi / tamsayı vb.) elemanlar içerebilirler.
- Bununla birlikte, küme elemanları "değiştirilemez (immutable)" türde olmak zorundadır.
- Küme elemanları "değiştirilemez (immutable)" türde olmak zorunda olsa da, kümelerin kendisi değiştirilebilir (mutable) türdedir.
- Sözlükler genişletilebilir veya daraltılabilir.
- Kümeleri diğer veri tiplerinden ayıran bir diğer fark, aynı elemanının bir kümede ancak bir kere yer alabilmesidir.
- Python'daki kümeler, matematikte kullanılan küme operatörleri (altküme, birleşim, kesişim, fark vb.) desteklediğinden oldukça kullanışlıdırlar.

Kümeler (Sets)

Kümelerin tanımlanması:

Kümeler doğrudan set() fonksiyonu ile veya {} operatörleri ile tanımlanabilir.

```
>>> iller = set(["Ankara", "Adana", "Samsun"])
>>> print(iller)
{'Ankara', 'Adana', 'Samsun'}
>>> iller = set({"Ankara":"06", "Adana": "01", "Samsun":"55"})
>>> print(iller)
{'Adana', 'Ankara', 'Samsun'}
>>> iller = {"Ankara", "Adana", "Samsun"}
>>> print(iller)
{'Adana', 'Ankara', 'Samsun'}
>>> x = set("JFM212")
>>> print(x)
{'M', 'J', '2', '1', 'F'}
```

Küme Fonksiyonları

add():

Küme elemanları değiştirilemez türde olmak zorundadır. Bu nedenle "add()" fonksiyonuna verilecek parametre de değiştirilemez türde olmak zorundadır.

```
>>> renkler = set(["Sarı", "Mavi", "Yeşil"])
>>> renkler.add('Kırmızı')
>>> print(renkler)
{'Yeşil', 'Mavi', 'Kırmızı', 'Sarı'}
>>> renkler.add(['Kırmızı','Turuncu'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'list'
```

Küme Fonksiyonları

update():

Mevcut bir kümeye "add()" fonksiyonu ile bir eleman eklenebilir. Birden fazla eleman eklenmek istenirse, "update()" komutu kullanılabilir.

```
>>> renkler = set(["Sarı", "Mavi", "Yeşil"])
>>> renkler.update(["Turuncu","Kırmızı"])
>>> print(renkler)
{'Yeşil', 'Mavi', 'Turuncu', 'Kırmızı', 'Sarı'}
```

Küme Fonksiyonları

clear():

Bir kümeden tüm elemanları siler. Ancak, kümenin kendisini silmez, sadece boş küme haline getirir.

```
>>> renkler = set(["Sarı", "Mavi", "Yeşil"])
>>> renkler.clear()
>>> print(renkler)
set()
```

Küme Fonksiyonları

copy():

Bir kümeden tüm elemanları siler. Ancak, kümenin kendisini silmez, sadece boş küme haline getirir.

```
>>> renkler = set(["Sarı", "Mavi", "Yeşil"])
>>> renkler2 = renkler
>>> renkler2.clear()
>>> print(renkler)
set()
>>> renkler = set(["Sarı", "Mavi", "Yeşil"])
>>> renkler2 = renkler.copy()
>>> renkler2.clear()
>>> print(renkler)
{'Yeşil', 'Mavi', 'Sarı'}
```

Küme Fonksiyonları

difference(): (-)

Bir küme ile diğer bir kümenin farkını verir (A \ B).

```
>>> x = {"a","b","c","d","e"}
>>> y = {"b","c"}
>>> z = {"c","d"}
>>> x.difference(y)
{'a', 'e', 'd'}
>>> x.difference(y).difference(z) {'a', 'e'}
```

Aynı işlem "-" operatörü ile de yapılabilir:

```
>>> x - y {'a', 'e', 'd'}
>>> x - y – z
{'a', 'e'}
```

Küme Fonksiyonları

symmetric_difference(): (^)

Bir küme ile diğer bir kümenin her iki küme için de farkını verir.

```
>>> x = {"a","b","c","d","e"}
>>> y = {"b","c"}
>>> z = {"c","d"}
>>> x ^ y
{'a', 'd', 'e'}
>>> y ^ z
{'b', 'd'}
>>> y - z
{'b'}
```

Küme Fonksiyonları

difference_update():

Bir küme ile diğer bir kümenin farkını (A \ B) alarak, A'yı sadece bu elemanlardan oluşacak hale getirir. (A = A \ B)

```
>>> x = {"a","b","c","d","e"}
>>> y = {"b","c"}
>>> x.difference_update(y)
>>> print(x)
{'d', 'a', 'e'}
```

Küme Fonksiyonları

discard(eleman):

Bir kümede elemanı kümeden çıkarır. Bu eleman kümede yoksa hata mesajı vermez.

```
>>> x = {"a","b","c","d","e"}
>>> x.discard("a")
>>> x {'c', 'b', 'e', 'd'}
>>> x.discard("z")
>>> x
{'c', 'b', 'e', 'd'}
```

Küme Fonksiyonları

remove(eleman):

Aynı "discard" fonksiyonu gibi olmakla birlikte, çıkarılacak eleman kümede yoksa hata mesajı verir.

```
>>> x = {"a","b","c","d","e"}
>>> x.remove("a")
>>> x
{'b', 'c', 'd', 'e'}
>>> x.remove("z")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'z'
```

Küme Fonksiyonları

intersection(küme): (&)

Bir kümenin diğer bir küme ile olan kesişim kümesini bulur.

```
>>> x = {"a","b","c","d","e"}
>>> y = {"c","d","e","f","g"}
>>> x.intersection(y)
{'c', 'e', 'd'}
```

VEYA

```
>>> x & y
{'c', 'e', 'd'}
```

Küme Fonksiyonları

isdisjoint()

Bir kümenin diğer bir küme ayrık kümeler olup olmadığını bulur.

```
>>> x = {"a","b","c","d","e"}
>>> y = {"c","d","e","f","g"}
>>> x.isdisjoint(y)
False
```

```
>>> x = {"a","b","c","d","e"}
>>> y = {"h","j","k","f","g"}
>>> x.isdisjoint(y)
True
```

Küme Fonksiyonları

issubset() (<=)

Bir kümenin diğer bir kümenin altkümesi olup olmadığını bulur.

```
>>> x = {"c","d"}
>>> y = {"a","b","c","d","e"}
>>> x.issubset(y)
True
```

```
>>> x = {"c","d"}
>>> y = {"a","b","c","d","e"}
>>> x <= y
True
```

Küme Fonksiyonları

issuperset() (>=)

Bir kümenin diğer bir kümenin altkümesi olup olmadığını bulur.

```
>>> x = {"c","d"}
>>> y = {"a","b","c","d","e"}
>>> y.issuperset(x)
True
```

```
>>> x = {"c","d"}
>>> y = {"a","b","c","d","e"}
>>> y >= x
True
```

Küme Fonksiyonları

pop()

Bu fonksiyon, kümenin rasgele bir elemanını verir ve bu elemanı kümeden çıkarır.

```
>>> x = {"a","b","c","d","e"}
>>> x.pop()
'b'
>>> print(x)
{'c', 'a', 'd', 'e'}
>>> x.pop()
'c'
>>> print(x)
{'a', 'd', 'e'}
```

Küme Fonksiyonları

Birden Fazla Küme:

Birleşim ve kesişim fonksiyonları birden fazla küme üzerinde çalışabilir.

```
>>> x = {"a","b","c","d","e"}
>>> y = {"h","j","c","d","k"}
>>> z = {"i","b","e","d","n"}
>>> set.intersection(x,y,z)
{'d'}
>>> set.union(x,y,z)
{'j', 'd', 'i', 'b', 'c', 'a', 'k', 'n', 'h', 'e'}
```

Küme Fonksiyonları

Küme elemanı olup olmadığını kontrol:

```
>>> x = {"a","b","c","d","e"}
>>> 'c' in x
True
>>> 'p' in x
False
```

Küme Fonksiyonları

Küme elemanları üzerinde döngüler:

```
>>> x = {"a","b","c","d","e"}
>>> for harf in x:
...   print(harf)
...
b
c
a
d
e
```

Küme Fonksiyonları

Dondurulmuş Kümeler:

"Dondurulmuş kümeler" normal kümeler ile aynı özelliklere sahip olmakla birlikte, kümelerden farklı olarak oluşturulduktan sonra değiştirilemezler. "Değiştirilemez (immutable)" tipte olmalarının birçok kullanım alanı vardır.

Örneğin, sözlüklerde anahtar olabilirler veya bir kümenin elemanı olabilirler

```
>>> x = frozenset(["a","b","c","d","e"])
>>> y = set([x,3,4,5])
>>> y
{3, 4, frozenset({'b', 'c', 'a', 'd', 'e'}), 5}
>>> x.add('f')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'frozenset' object has no attribute 'add'
```

Kaynakça

1 Wentworth, P., Elkner, J., Downey, A.B., Meyers, C. (2014). How to Think Like a Computer Scientist: Learning with Python (3nd edition).
2 Pilgrim, M. (2014). Dive into Python 3 by. Free online version: DiveIntoPython3.org ISBN: 978-1430224150.
3 Summerfield, M. (2014) Programming in Python 3 2nd ed (PIP3) : - Addison Wesley ISBN: 0-321-68056-1.
4 Summerfield, M. (2014) Programming in Python 3 2nd ed (PIP3) : - Addison Wesley ISBN: 0-321-68056-1.
5 Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/.
6 Millman, K.J., Aivazis, M. (2011). Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12.
7 John D. Hunter (2007). Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95.
8 Travis E. Oliphant (2007). Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20.
9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). Data Structures and Algorithms in Python, Wiley.
10 http://www.diveintopython.net/
11 https://docs.python.org/3/tutorial/
12 http://www.python-course.eu
13 https://developers.google.com/edu/python/
14 http://learnpythonthehardway.org/book/