

PYTHON PROGRAMLAMA DİLİ

VERİ TİPLERİ
set

Dr. Öğr. Üyesi M. Ozan AKI

VERİ TİPLERİ

Python dilinde, diğer programlama dillerinde olduğu gibi farklı veri tipleri bulunmaktadır.

int, **float** ve **str** sadece bir değişken tanımlarken,

List, **Tuple**, **Dictionary** ve **Set** veri tipleri, çok sayıda veriyi sistematik bir şekilde saklamak ve işlemek için kullanılmaktadır.

Veri Tipi	Açıklama
int	Tam sayı 123
float	Ondalıklı sayılar 12.34
str	Karakter dizileri '...' “...”
List	Listeler [...]
Tuple	Demetler (...)
Dictionary	Sözlük {...}
Set	Küme {...}

SET (KÜME)

Kümeler, matematikteki kümelerdir.

Kümenin her bir elemanı benzersizdir.

Yani her elemandan sadece bir tane bulunabilir.

Kümeler arasında kesişim, birleşim, fark kümeleri gibi işlemler yapılabilir.

Kümeler, {...} küme parantezi ile tanımlanır.

`set.add()`

`set.remove()`

`set.discard()`

`set.update()`

`set.pop()`

`set.clear()`

`set.copy()`

`set.union()`

`set.intersection()`

`set.intersection_update()`

`set.difference()`

`set.difference_update()`

`set.symmetric_difference()`

`set.symmetric_difference_update()`

```
>>> liste='ankara'
>>> print(liste)
ankara
>>> kume=set(liste)
>>> print(kume)
{'n', 'a', 'k', 'r'}
>>> |
```

SET (KÜME) - METOTLAR

set.add(e): Kümeye **e** ile verilen elemanı (yoksa) ekler.

set.remove(e): Kümeden **e** ile verilen elemanı siler. Eleman yoksa hata verir.

set.discard(e): Kümeden **e** ile verilen elemanı siler. Eleman yoksa hata vermez.

set.update(e): Kümeye, **e** ile verilen eleman listesini topluca ekler.

set.copy(): Kümenin bir kopyasını döndürür.

set.clear(): Kümenin tüm elemanlarını siler, boş küme olur.

```
>>> asal={2,3,5,7,11,13}
>>> type(asal)
<class 'set'>
>>> print(asal)
{2, 3, 5, 7, 11, 13}
>>> asal.add(15)
>>> print(asal)
{2, 3, 5, 7, 11, 13, 15}
>>> asal.remove(15)
>>> print(asal)
{2, 3, 5, 7, 11, 13}
>>> asal.update([17,23,27])
>>> print(asal)
{2, 3, 5, 7, 11, 13, 17, 23, 27}
>>> asal.clear()
>>> print(asal)
set()
>>>
```

SET (KÜME) - METOTLAR

A.union(B) ($A \cup B$): **A** ile **B** kümesinin bileşim kümesini döndürür.

A.update(B) ($A \cup B$): **A** ile **B** kümesinin bileşim kümesini **A** kümesine atar.

A.intersection(B) ($A \cap B$): **A** ile **B** kümesinin kesişim kümesini döndürür.

A.intersection_update(B) ($A \cap B$): **A** ile **B** kümesinin kesişim kümesini **A** kümesine atar.

```
>>> asal={2,3,5,7,11,13}
>>> cift={2,4,6,8,10,12}
>>> asal.union(cift)
{2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13}
>>> asal|cift
{2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13}
>>> asal.intersection(cift)
{2}
>>> asal&cift
{2}
>>>
```

SET (KÜME) - METOTLAR

A.difference(B) ($A - B$): **A** fark **B** kümesini döndürür.

A.difference_update(B) ($A -= B$): **A** fark **B** kümesini, **A** kümesine atar.

A.symmetric_difference(B) ($A \oplus B$): **A** ile **B** kümesindeki, birbirinde olmayan elemanların kümesini döndürür.

A.symmetric_difference_update(B) ($A \oplus= B$): **A** ile **B** kümesindeki, birbirinde olmayan elemanların kümesini **A** kümesine atar.

```
>>> asal={2,3,5,7,11,13}
>>> cift={2,4,6,8,10,12}
>>> asal.difference(cift)
{3, 5, 7, 11, 13}
>>> asal-cift
{3, 5, 7, 11, 13}
>>> cift.difference(asal)
{4, 6, 8, 10, 12}
>>> cift-asal
{4, 6, 8, 10, 12}
>>> asal.symmetric_difference(cift)
{3, 4, 5, 6, 7, 8, 10, 11, 12, 13}
>>> asal^cift
{3, 4, 5, 6, 7, 8, 10, 11, 12, 13}
>>>
```

SET (KÜME) - METOTLAR

A.isdisjoint(B): A ile B kümesinin kesişim kümesi boş küme ise True, değilse False döndürür.

A.issubset(B): A kümesi, B kümesinin bir alt kümesi ise True, değilse False döndürür.

A.issuperset(B): A kümesi, B kümesini kapsıyorsa True, kapsamıyorsa False döndürür.

```
>>> tekler={1,3,5,7,9}
>>> ciftler={2,4,6,8,10}
>>> tekler.isdisjoint(ciftler)
True

>>>
>>> sinif={'ali','banu','can','didem','efe'}
>>> grup={'can','efe'}
>>> grup.issubset(sinif)
True
>>> sinif.issuperset(grup)
True
>>> |
```