

PYTHON PROGRAMLAMA DİLİ

VERİ TİPLERİ
list

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 {...}

LIST (LİSTELER)

Birbiriyle ilişkili verilerin tek bir isim altında saklanması ve işlenmesini sağlayan yapılardır.

Listeler, [...] köşeli parantez ile tanımlanır. Köşeli parantez içinde elemanlar tanımlanır.

Not: Listelerdeki elemanların C/C++ dilinde olduğu gibi aynı tipte olması zorunluluğu yoktur. Listenin elemanları yine bir liste olabilir.

```
>>> sayilar=[1,2,3,4,5]
>>> oranlar=[1.2,3.14,5.25,8.25]
>>> harfler=['a','b','c','d','e']
>>> isimler=['ali','beyza','can','didem']
>>> karisik=[5,'ali',3.14,'x','mustafa',123,'mavi',0.005]
>>> icice=['abc',5,[3,'mehmet',5],'xyz',2.71,1234]
>>> print(icice[1])
5
>>> print(icice[2])
[3, 'mehmet', 5]
>>> |
```

LIST (LİSTELER) - INDEX

Listelerin elemanlarına yine [] köşeli parantez içerisine indis numarası yazılarak erişilir.

İndisler 0'dan (sıfır) başlar. İlk elemanın indisi "0" dır.

Sondan başa doğru ilerlemek için negatif indisler kullanılır. Son elemanın indisi "-1" dir.

Liste elemanlarında okuma, değiştirme ya da yenisini yazma işlemleri yapılabilir.

```
>>> dizi=[12,23,34,45,56,67]
>>> print(dizi)
[12, 23, 34, 45, 56, 67]
>>> print(dizi[2])
34
>>> dizi[1]=99
>>> print(dizi)
[12, 99, 34, 45, 56, 67]
>>> dizi[2]+=6
>>> print(dizi)
[12, 99, 40, 45, 56, 67]
>>> dizi[3]='mehmet'
>>> print(dizi)
[12, 99, 40, 'mehmet', 56, 67]
>>> print(dizi[-1])
67
>>>
```

LIST (LİSTELER) - SLICE

Slice işlemi ile listelerden “dilimler” kesilerek alınabilir.

Bunu için : (iki nokta üst üste) sembolü ile bir aralık tanımlanır.

Örneğin, listenin 3. elemanından 8. elemanına kadar kısmını kesi almak istersek;

liste[3:8]

İstenirse adım sayısı da belirtilebilir. Örneğin, 1. elemandan 9. elemana kadar ikişer atlayarak elemanlara ulaşmak istenirse;

liste[1:9:2]

Dönen değerler yine bir liste olacaktır.

```
>>> liste=[0,1,2,3,4,5,6,7,8,9,10]
>>> parca1=liste[2:5]
>>> print(parca1)
[2, 3, 4]
>>> parca2=liste[7:10]
>>> print(parca2)
[7, 8, 9]
>>> parca3=liste[1:9:2]
>>> print(parca3)
[1, 3, 5, 7]
>>> parca4=liste[8:2:-1]
>>> print(parca4)
[8, 7, 6, 5, 4, 3]
>>> parca5=liste[9:1:-2]
>>> print(parca5)
[9, 7, 5, 3]
>>> |
```

LIST (LİSTELER) - SLICE

Eğer “:” sembolünden önce rakam yazılmazsa bu “Listenin başından” anlamına gelir.

Örneğin, listenin başından 5. elemana kadar kesmek için;

```
liste[:5]
```

Eğer “:” sembolünden sonra rakam yazılmazsa bu “Listenin sonuna kadar” anlamına gelir.

Örneğin, 7. elemandan listenin sonuna kadar kesmek için;

```
liste[7:]
```

Eğer sadece “:” kullanılırsa bu “Listenin başından sonuna kadar” anlamına gelir.

```
liste[:]
```

```
>>> liste=[0,1,2,3,4,5,6,7,8,9,10]
>>> parca6=liste[:5]
>>> print(parca6)
[0, 1, 2, 3, 4]
>>> parca7=liste[7:]
>>> print(parca7)
[7, 8, 9, 10]
>>> parca8=liste[:]
>>> print(parca8)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> |
```

LIST (LİSTELER) – FONKSİYON VE OPERATÖRLER

Listelerin üzerinde işlem yapan bazı fonksiyon ve operatörler bulunur.

len(liste)

Listenin eleman sayısını döndürür.
Dönen değer bir tam sayıdır

in operatörü, bir elemanın listede olup olmadığını döndürür. Dönen bir bool tipinde değerdir.

“Ahmet” in liste

“Ahmet” isminin listede olup olamamasına bağlı olarak True ya da False döner.

+= Operatörü, bir listenin sonuna başka bir listeyi eklemek için kullanılır.

```
>>> liste=['ankara','bursa','canakkale','denizli']
>>> len(liste)
4
>>> 'bursa' in liste
True
>>> 'edirne' in liste
False
>>>
>>> notlar=[55,86,78,45,86,64,75]
>>> len(notlar)
7
>>> 70 in notlar
False
>>> 45 in notlar
True
>>> |
>>> liste=['ali','banu','can']
>>> liste+=['didem','efe','filiz']
>>> print(liste)
['ali', 'banu', 'can', 'didem', 'efe', 'filiz']
>>> |
```

LIST (LİSTELER) - METOTLAR

Listelerin, liste elemanları üzerinde çeşitli işlemler yapan “Metotları” vardır.

Listenin metotlarına, liste adından sonra bir . (nokta) konulduktan sonra erişilir.

Metotlar parantez içeirisinde parameter alabilirler.

Metotlar, bir değışkene atayabileceğiniz bir değer döndürebilirler.

```
list.append()  
list.extend()  
list.insert()  
list.remove()  
list.pop()  
list.clear()  
list.index()  
list.count()  
list.sort()  
list.reverse()  
list.copy()  
list.del()
```

LIST (LİSTELER) - METOTLAR

list.append(e): Listenin sonuna, parantez içinde verilen **e** elemanını ekler.

list.extend(v): Listenin sonuna **v** ile verilen başka bir liste ekler.

```
>>> liste=['ankara','bursa','canakkale','denizli']
>>> liste.append('edirne')
>>> print(liste)
['ankara', 'bursa', 'canakkale', 'denizli', 'edirne']
>>>
>>> notlar=[55,86,78,45,86,64,75]
>>> notlar.append(90)
>>> print(notlar)
[55, 86, 78, 45, 86, 64, 75, 90]
>>>
>>> liste=['Ali','Beril','Can']
>>> eklenecekler=['Didem','Efe','Filiz']
>>> print(liste)
['Ali', 'Beril', 'Can']
>>> liste.extend(eklenecekler)
>>> print(liste)
['Ali', 'Beril', 'Can', 'Didem', 'Efe', 'Filiz']
```

LIST (LİSTELER) - METOTLAR

list.insert(i,e): Listeye, **i** ile belirtilen indise **e** ile verilen elemanı ekler. Diğer elemanlar bir sağa kayar.

list.remove(e): Listedен **e** ile belirtilen elemanı siler.

list.pop(): Parametre verilmezse, sonunca elemanı döndürür ve listeden siler.

list.pop(i): Verilen **i** indisindeki elemanı döndürür ve onu listeden siler.

list.clear(): Tüm elemanları silerek listeyi boşaltır. Boş bir liste haline gelir.

```
>>> liste.insert(2, 'Cansu')
>>> print(liste)
['Ali', 'Beril', 'Cansu', 'Can', 'Didem', 'Efe', 'Filiz']
>>> liste.remove('Can')
>>> print(liste)
['Ali', 'Beril', 'Cansu', 'Didem', 'Efe', 'Filiz']
>>> liste.count('Didem')
1
>>> liste.pop()
'Filiz'
>>> print(liste)
['Ali', 'Beril', 'Cansu', 'Didem', 'Efe']
>>> liste.pop(1)
'Beril'
>>> print(liste)
['Ali', 'Cansu', 'Didem', 'Efe']
>>> liste.clear()
>>> print(liste)
[]
>>> |
```

LIST (LİSTELER) - METOTLAR

list.index(e): Verilen **e** elemanının liste içindeki indis numarasını döndürür.

list.count(e): Verilen **e** elemanının liste içinde kaç kez bulunduğunu sayarak döndürür.

list.sort(): Liste elemanlarını küçükten büyüğe sıralar.

list.sort(reverse=True): Liste elemanlarını büyükten küçüğe sıralar.

list.sort(key=k): Liste elemanlarını verilen **k** koşula göre küçükten büyüğe sıralar.

list.reverse(): Liste elemanlarının sıralamasını tersine çevirir.

list.copy(): Listenin bir kopyasını döndürür.

del list[i]: listedeki **i** indisindeki elemanı siler.

```
>>> notlar=[34,78,65,55,20,70,85,90,45]
>>> print(notlar)
[34, 78, 65, 55, 20, 70, 85, 90, 45]
>>> notlar.sort()
>>> print(notlar)
[20, 34, 45, 55, 65, 70, 78, 85, 90]
>>> notlar.reverse()
>>> print(notlar)
[90, 85, 78, 70, 65, 55, 45, 34, 20]
>>> |
```

```
>>> liste=['van','edirne','bursa','diyarbakir']
>>> print(liste)
['van', 'edirne', 'bursa', 'diyarbakir']
>>> liste.sort()
>>> print(liste)
['bursa', 'diyarbakir', 'edirne', 'van']
>>> liste.sort(reverse=True)
>>> print(liste)
['van', 'edirne', 'diyarbakir', 'bursa']
>>> liste.sort(key=len)
>>> print(liste)
['van', 'bursa', 'edirne', 'diyarbakir']
>>> liste.sort(key=len, reverse=True)
>>> print(liste)
['diyarbakir', 'edirne', 'bursa', 'van']
>>> |
```