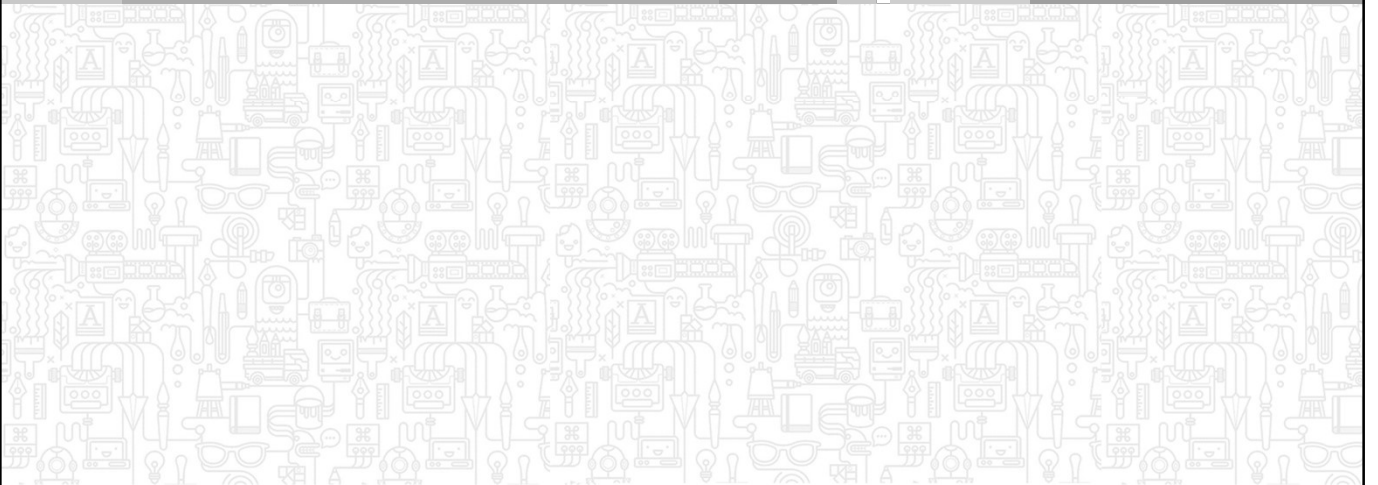


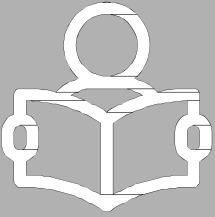
## PYTHON



## Listeler ve Özellikleri

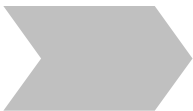


## Kapsam



- ✓ Liste Veri Tipleri
  - Liste Kavramı ve İndis Değerleri
  - Liste Elemanlarına Erişim
  - Temel Liste Metotları
  - len() Fonksiyonu ile Uzunluk Bilgisi
  - İç İçe Listeler
  - Veri Tipi Dönüşümleri

## Listeler



Daha önce bir değişken üzerinde sadece bir değer tutuyorduk. Listeler, bir değişkenin altında birden fazla değer tutulabilmesine yarar.



Listeler, içinde farklı türlerden verileri barındırabilen taşıyıcılar olarak adlandırılmaktadır. Listeler zaten Python'daki veri tiplerinden biridir.



İki köşeli parantez [ ] arasında tanımlanırlar



Listeler dilimlenebilir ve elemanları sonradan değiştirilebilir.

## Liste Veri Tipleri

Bir listede her veri tipinden eleman saklanabilir.

### Örnek 1

Kod

```
liste=[1,2,'ali',0.25]
print(liste)
```

int, string ve float gibi farklı veri tiplerini içerisinde barındıran 4 elemanlı bir liste

## Liste Kavramı ve İndis Değerleri



Öncelikle liste üzerinden veri okuyabilmek için hangi indis elemanının okunmak istendiği doğru bir şekilde belirtilmelidir.



Listelerde ilk eleman her zaman 0. indistir.  
Listenin son elemanı ise liste uzunluğun bir eksiğidir.



İndis numarası	Liste[0]	Liste[1]	Liste[2]	Liste[3]
Öge sıralaması	Birinci öge	İkinci öge	Üçüncü öge	Dördüncü öge
Liste =	'ali',	[ 5,	3.14,	'ayşe' ]

Listenin ilk elemanına erişmek veya yazmak istendiğinde listeAdı[0] yazılması gerekmektedir. Diğer elemanları için de sırasıyla listeAdı[1], listeAdı[2] ... şeklinde yazılması gerekmektedir.

## Liste Kavramı ve İndis Değerleri

Üç elemanlı bir liste ve indisleri

### Örnek 2

Kod

```
liste=["Ali", 3, 4.5]
print(liste)
print(liste[0])
print(liste[1])
print(liste[2])
```

## Liste Elemanlarına Erişim

- Listedeki öğelere ulaşmak için bir değişkene, listenin indis numarasına göre atama yapıp çağrılabilir.

- `degiskenAdı = liste[indis]`
  - `liste=["a",4,7,3.4]`
  - `listeOge=liste[1]`

- Liste oluştururken içine herhangi bir değer girilmeden de oluşturulup, sonradan değer ataması yapılabilir.

- `liste=[ ]`
- `liste[0]=4`
- `liste[1]=7`

- Herhangi bir indis numarasından başlanıp belirlenen indis numarasına kadar olan öğelere ulaşılır.
  - `dilim = liste[baslangic:bitis]`
- Listenin belirli aralıktaki öğelerini alma işlemine dilimleme denir.
- Liste dilimlerken adımlamada başlangıç indisi alır ama bitiş indisi almaz.
  - `liste=[0,1,2,3,4,5]`
  - `print(liste[0:4])` # çıktısı [0, 1, 2, 3]

### Örnek 3

#### Bazı örnekler

Kod

```
liste = [0,1,2,3,4,5,6,7,8,9,10]
print(liste) #[0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(liste[0]) #0
print(liste[5]) #5
print(liste[:5]) #[0,1, 2, 3, 4]
print(liste[1:7]) #[1, 2, 3, 4, 5, 6]
print(liste[5:]) #[5, 6, 7, 8, 9, 10]
print(liste[::2]) #[0, 2, 4, 6, 8, 10]
```

### Örnek 4

Liste içindeki öğeleri kontrol edilebilir.  
Eşya adında bir liste var. Liste tanımlanır, daha sonra if karar yapısı ile içindeki öğeler de “perde” olup olmadığı kontrol edilir.

Kod

```
esya = ["ayna", "televizyon", "perde"]
if("perde" in esya):
    print("Bu değer listede var.")
else:
    print("Bu değer listede yok.")
```

## Örnek 5

Aşağıdaki örnekte 5 elemanlı bir liste tanımlanmıştır. İndis numaralarını kullanarak, listenin 1. indisten başlayarak, 3. indise kadar öge değişkenine aktarıp listeleme işlemi yapılmıştır.

Kod

```
liste = [1,2,3,4,5]
öge = listem[1:3]
print(öge)
```

## Negatif İndisler

Negatif indis numarası listenin sonuncu elemanından başlayarak sayıldığında (sondan başa) sıra numarasını verir.

## Örnek 6

Kod

```
liste = ["merhaba","iyi günler","nasılsın","güle güle"]
print(liste[- 1]) #son ögeyi listeler
print(liste[- 3]) #sondan üçüncü ögeyi listeler
print(liste[- 4]) #sondan dördüncü ögeyi listeler
print(liste[::-1]) #sondan başa doğru listeleme yapmak için kullanılır
```

Örnek'te listenin sonuncu elemanına liste[-1] olarak, sondan ikinciye ise [-2] olarak ulaşılır. "güle güle" elemanı listede -1. indis olarak görülmektedir. "merhaba" ise geriye doğru sayıldığında -4. indis'tir.

## Temel Liste Metotları

Metotlar listelerin işlevlerine erişilmesini sağlar. Listenin metotları için `dir(list)` ile tüm metotlar görülebilir. Bu metotlar yardımıyla listeler ekleme, çıkarma, arama, sıralama vb. birçok işlemin kolaylıkla yapılabilmesini sağlamaktadır.

Sıra No	Metot Adı	Görevi
1	'append'	Listeye yeni eleman ekleme işlemini yapar. Bu metot ile listeye sadece bir eleman eklenebilir ve eklenen eleman listenin sonunda yer alır.
2	'clear'	Listeyi değil içindeki tüm ifadeleri silmeye yarar.
3	'copy'	Listeden listeye kopyalama işlevine yaramaktadır.
4	'count'	Listenin içinde sorgulanan elemandan kaç adet olduğunu bulmamızı sağlar.
5	'extend'	Listeler arası genişletme işlevini görür.
6	'index'	Listedeki elemanları almamızı sağlar.
7	'insert'	Listenin istenilen indis numarasına eleman eklenebilir.
8	'pop'	Listedeki elemanın indisi ile silme işlemi yapar. indis belirtmediğinizde ise varsayılan olarak listenin son elemanını siler. Ayrıca bu metot silinen elemanı ekrana yazmaktadır.
9	'remove'	Listede istenilen elemanın değerini yazarak silme işlemi yapar.
10	'sort'	Listenin elemanlarını alfabetik olarak sıralar.
11	'reverse'	Bu metot sort metodunun aksine listedeki elemanları ters alfabetik olarak sıralar.
12	'del'	Liste içerisinde bir elemanı silmek için kullanılır. Silme işlemi indis numarasına göre yapılmaktadır.

## 'append' kullanımı

Listeye yeni eleman ekleme işlemini yapar. Bu metot ile listeye sadece bir eleman eklenebilir ve eklenen eleman listenin sonunda yer alır.

Kod

```
takimlar=["gs","fb","bjk"]
takimlar.append("ts")
print(takimlar)
```

takimlar listesine "ts" ögesi eklenmiş ve son indis numarasında yer almıştır.

## 'insert' kullanımı

Listenin istenilen indis numarasına eleman eklenebilir.

Kod

```
sebzeler = ["lahana", "marul", "pırasa", "ıspanak", "fasulye"]  
sebzeler.insert(2, "patlıcan")  
print(sebzeler)
```

2. indis numarasına "patlıcan" ögesi eklenmiştir.

## 'copy' kullanımı

Listeden listeye kopyalama işlevine yaramaktadır.

Kod

```
iller1 = ["konya", "karaman", "kocaeli", "kayseri", "kahramanmaraş"]  
iller2 = []  
iller2 = iller1.copy()  
print(iller2)
```

iller1 listesi copy metodu ile iller2 listesine aktarılmıştır.



## 'count' kullanımı

Listenin içinde sorgulanan elemandan kaç adet olduğunu bulmamızı sağlar.

Kod

```
takimlar = ['GS', 'FB', 'BJK', 'TS']  
print(takimlar.count('FB'))
```

takimlar listesinde 'FB' ögesinin kaç adet olduğu count metodu bulunmuştur.

## 'extend' kullanımı

Listeler arası genişletme işlevini görür.

Kod

```
kus1=["bıldırcın", "papağan", "kartal", "akbaba", "şahin"]  
kus2=["baykuş", "muhabbet"]  
kus1.extend(kus2)  
print(kus1)
```

Listelerdeki öğelerin kendi elemanlarını koruyarak genişletme işlemi yapılmıştır.

## 'index' kullanımı

Listedeki elemanları almamızı sağlar.

Kod

```
sebzeler = ["lahana", "marul", "pırasa", "ıspanak", "fasulye"]  
print(sebzeler.index("ıspanak"))
```

index metodu görüldüğü gibi verilen bir ögenin indis numarasını vermektedir.

## 'clear' kullanımı

Listeyi değil içindeki tüm ifadeleri silmeye yarar.

Kod

```
liste = ["ayva", "nar", "kiraz", "kayısı", "üzüm"]  
liste.clear()  
print(liste)
```

Listenin tüm öğeleri silinmiştir.

## 'pop' kullanımı

Listedeki elemanın indisi ile silme işlem yapar. indisi belirtmediğinizde ise varsayılan olarak listenin son elemanını siler. Ayrıca bu metod silinen elemanı ekrana yazmaktadır.

Kod

```
sebzeler = ["lahana", "marul", "pırasa", "ıspanak", "fasulye"]  
sebzeler.pop(2)  
print(sebzeler)
```

sebzeler listesinden 2.indis numarasına ait olan "pırasa" adlı öge silinmiştir.

## 'remove' kullanımı

Listede istenilen elemanın değerini yazarak silme işlemi yarar.

Kod

```
sehirler = ["adana", "ağrı", "bursa", "konya", "ankara"]  
sehirler.remove("konya")  
print(sehirler)
```

öge adına göre silme işlemi yapılmıştır ve "konya" adlı öge sehirler listesinden silinmiştir.

## 'reverse' kullanımı

Bu metot sort metodunun aksine listedeki elemanları ters alfabetik olarak sıralar.

Kod

```
sayilar=[10,20,30,40,50,60,70]
sayilar.reverse()
print(sayilar)
```

liste öge elemanları tersten sıralanmıştır.

## 'sort' kullanımı

Listenin elemanlarını sıralar.

Kod

```
isimler=["elif","ayşe","kemal","kaan","hafsa"]
isimler.sort()
print(isimler)
liste=[3,1,5,89]
liste.sort()
liste
```

isimler listesi öğeleri alfabetik olarak sıralanmıştır, küçükten büyüğe doğru sıralanmıştır.

## 'del' kullanımı

Liste içerisinde bir elemanı silmek için kullanılır. Silme işlemi indis numarasına göre yapılmaktadır.

Kod

```
takimlar = ['GS', 'FB', 'BJK', 'TS']
del takimlar[2]
print(takimlar)
```

takimlar listesine ait 'BJK' ögesi indis numarasına göre silinmiştir.

## len() Fonksiyonu ile Uzunluk Bilgisi

len( ) fonksiyonu, string ifadesinin uzunluğunu yani karakter sayısını verir.

Kod

```
a="Galatasaray"
print(len(a))
takimlar = ['GS', 'FB', 'BJK', 'TS']
print( len(takimlar))
```

len( ) kullanımına bakıldığında, a ismindeki değişkenin uzunluğu 8 ve takimlar adındaki listenin eleman sayısı ise 4 olarak verilmiştir.

## İç İçe Listeler

Bir liste herhangi bir sıralama nesnesi içerebilir, hatta başka bir liste (alt liste) içerebilir, alt listeler de alt listeler içerebilir ve bu şekilde devam eder. Bu yuvalanmış liste olarak bilinir. Hiyerarşik yapılara veri düzenlemek için bunlar kullanılabilir.

Kod

```
liste1 = [1,2,3]
liste2 = [4,5,6]
liste3 = [7,8,9]
yeniListe = [liste1,liste2,liste3]
print(yeniListe)
```

Bu listelerin her biri ayrı öge olacak şekilde birleştirilerek iç içe liste oluşturuldu ve ekrana yazdırıldı.

yeniListe; [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

## İç İçe Listeler

append () metodu ile iç içe listeler oluşturulabilir.

Kod

```
sebzeler=[]
sebzeler.append(['yeşil','ıspanak'])
sebzeler.append(['beyaz','lahana'])
sebzeler.append(['turuncu','havuç'])
sebzeler.append(['siyah','turp'])
sebzeler.append(['kırmızı','domates'])
print(sebzeler)
print(sebzeler[0]) #['yeşil', 'ıspanak']
print(sebzeler[2][1]) #havuç
```

append ( ) metodunu kullanarak sebzeler adlı listeye 5 adet eleman girildi. Girilen verilerin listelenmesi: [['yeşil', 'ıspanak'], ['beyaz', 'lahana'], ['turuncu', 'havuç'], ['siyah', 'turp'], ['kırmızı', 'domates']]

Liste adı	sebzeler				
Eleman değeri	'yeşil', 'ıspanak'	'beyaz', 'lahana'	'turuncu', 'havuç'	'siyah', 'turp'	'kırmızı', 'domates'
İndis numarası	0	1	2	3	4

## İç İçe Listeler

Örnek'te 4 adet liste oluşturulmuştur. İlk üç listenin tüm öğeleri son\_liste adında listeye aktarılarak, ekrana yazdırılmıştır.

Kod

```
birinci_liste = [1,2,3]
ikinci_liste = ['a','b','c']
ucuncu_liste= [40,50,60]
son_liste= [birinci_liste,ikinci_liste,ucuncu_liste]
print(son_liste)
print(son_liste[1][0],son_liste[2][1]) #a50
```

```
[[1, 2, 3], ['a', 'b', 'c'], [40, 50, 60]]
```

## Listelerde Atamalar

Listelerde elemanlara yeni değer ataması yapılırken string, int, float vb. veri tipi tanımlaması yapılabilir, yapılmayabilir de, Python tanımlamayı kendisi yapacaktır.

Kod

```
liste=[1,2,3,4,5,'ankara']
print(liste)
liste[0]=str("kocaeli")
liste[2]=float(1.5)
liste[5]=int(20)
print(liste)
```

## Veri Tipi Dönüşümleri

Herhangi bir string türünde veriyi parçalayarak da liste oluşturulabilir. Örnek'te string veri türünde adı meyve olan bir değişken tanımlanmış ve veri olarak "elma" atanmıştır. Meyve adlı değişken liste yardımıyla parçalanmıştır.

Kod

```
meyve="elma"  
liste=list(meyve)  
print (liste)
```

['e', 'l', 'm', 'a']

## Veri Tipi Dönüşümleri

Örnek'te 1 ile 15 arasındaki sayılardan oluşan liste oluşturulmuştur. Listenin öğeleri ekrana listelenmiştir. Ekrana listeleme işleminde sort() metodu ile öğeler küçükten büyüğe, reverse ile büyükten küçüğe doğru sıralanmıştır.

Kod

```
liste=list(range(1,15,2))  
print(liste)  
liste.sort()  
print(liste)  
liste.reverse()  
print(liste)
```



## split( ) metodu

split( ) karakter dizilerini istenen şekilde böler ve liste tanımlar. -ayırıcı diye tanımladığımız ilk parametre, karakter dizisinin nereden bölüneceğini seçer. Eğer ayırıcı tanımlanmazsa karakter dizisi her boşluk gördüğünde ayırır.

Ayrıca bir karakter dizisini listeye dönüştürür. list()'den farklı olarak belirtilen ayırıcıya kadar alır.

Kod

```
bilgi=input("bilgilerinizi araya virgül koyarak yazınız: ")
liste=bilgi.split(",")
print(liste)

meyve="elma"
liste=meyve.split()
print(liste) #['elma']
liste=meyve.split("m")
print(liste) #['el', 'a']

cumle="23 nisan herkese kutlu olsun"
kelimeler=cumle.split()
print(kelimeler) #['23', 'nisan', 'herkese', 'kutlu', 'olsun']
```