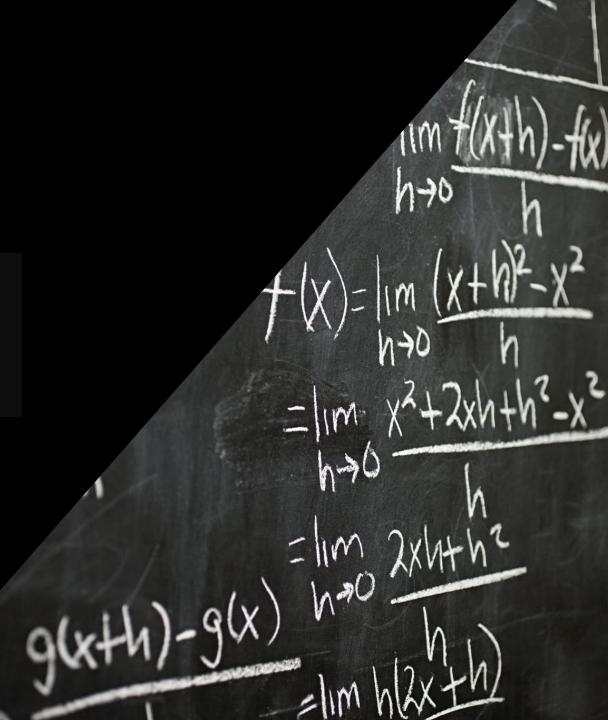
# PROGRAMLAMA TEMELLERI DERS 2

> Listeler

Python'da listeler, birden çok öğeyi saklamak için kullanılan veri yapılarıdır. Farklı verilerin bir dizi hâlinde tutulduğu koleksiyonlara liste adı verilir. Listeler, çeşitli veri tiplerini içerebilir ve değiştirilebilirler. Python programlama dilinde listeler iki köşeli parantez ile tanımlanmaktadır.

### \* KULLANIMI

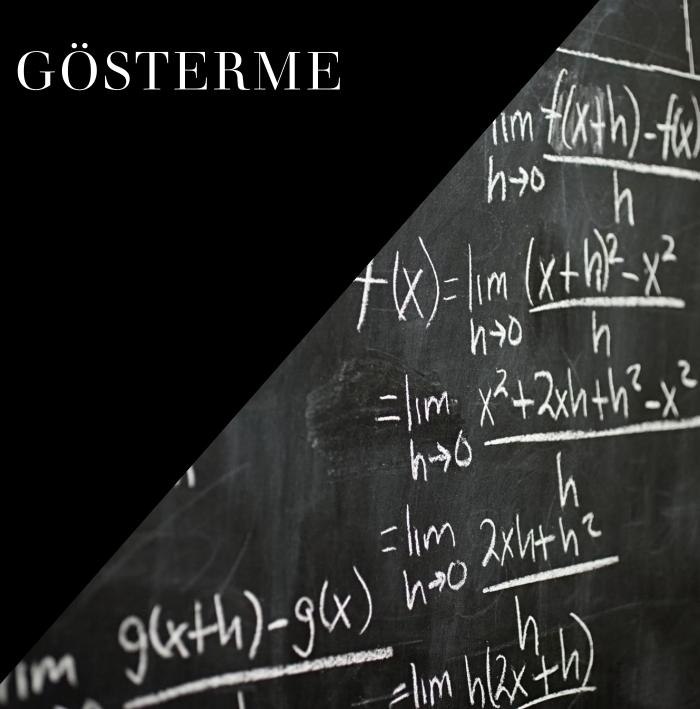
```
my_list = [1, 2, 3, 4, 5]
empty_list = []
mixed_list = [1, "Hello", 3.14, True]
```



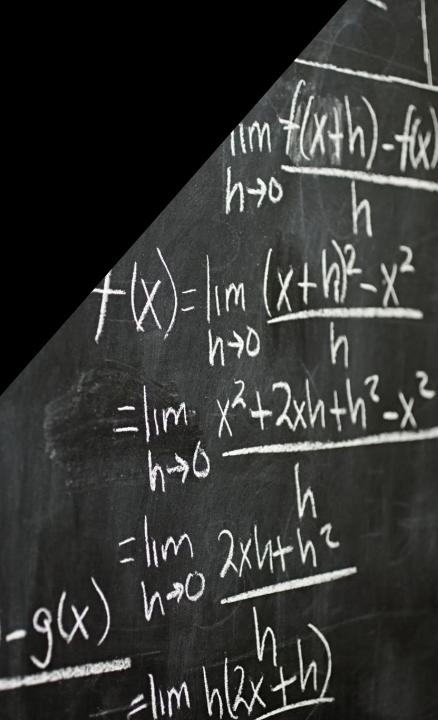
## > TYPE – TÜRÜNÜ GÖSTERME

```
my_list = [1, 2, 3, 4, 5]
list_type = type(my_list)
print(list_type)
```

<class 'list'>



> Örnek: Elemanları haftanın günleri olan bir liste oluşturunuz ve ekrana yazdırınız.



# Hoftonin günleri ["pazartesi", "Sall", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "pazar"]

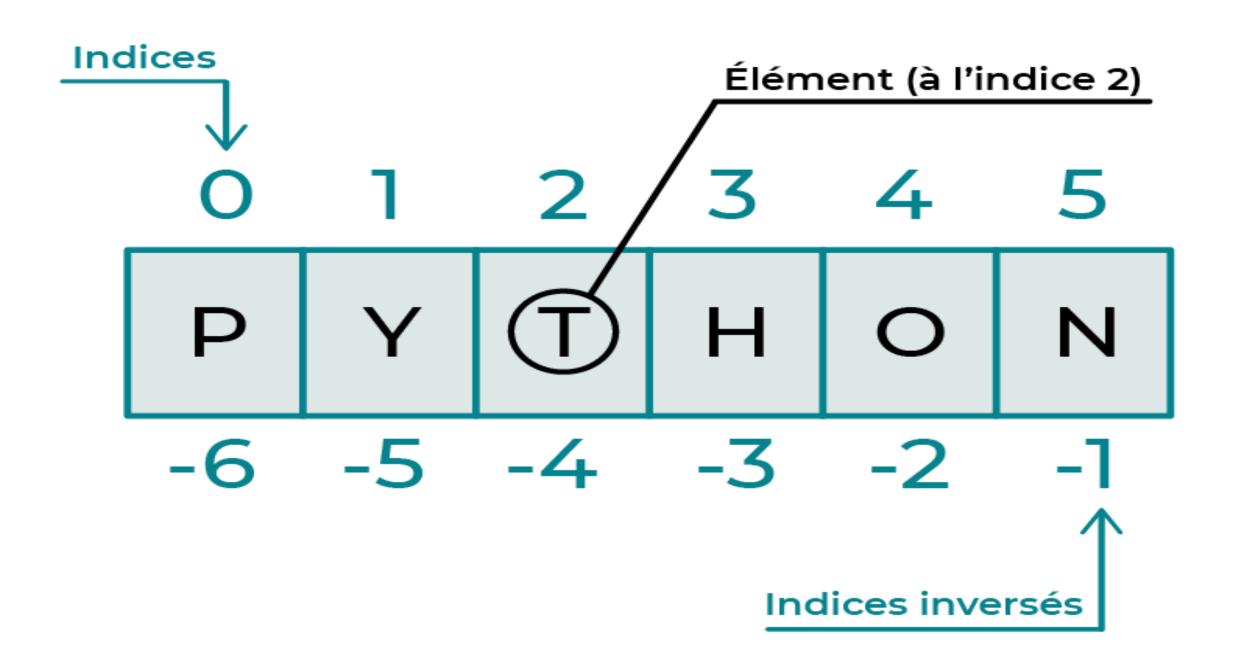
# Hoftonin günleri ["pazartesi", "Sall", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "pazar"] CEVAP # Listeyi doğrudan ekrana yazdırma haftanin\_gunleri)

# Listeyi doğrudan Günleri:", haftanin\_gunleri) N-30 XXVITh2

### > INDEKS KULLANIMI

Liste içindeki elemanlara erişmek için ilgili elemanın indeksi kullanılır. Bazı kaynaklarda indis olarak da karşınıza çıkabilir. İlk elemanın indisi her zaman D (sıfır) olarak kabul edilir.

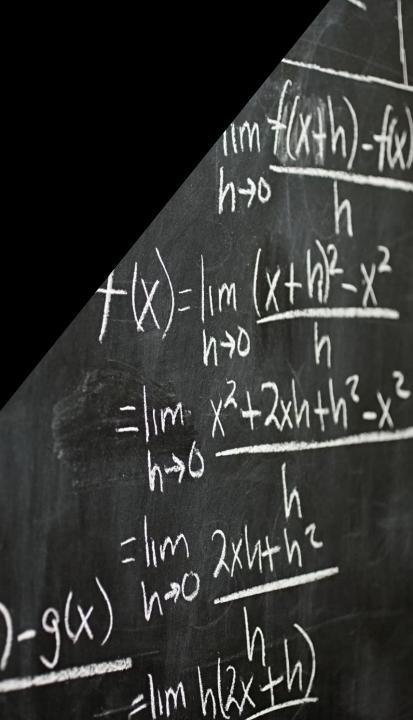
sehirler=["Ankara", "Bursa", "Çanakkale", "Denizli", "Eskişehir"] Şehirler isimli listenin ilk elemanı olan "Ankara", indeksi sıfır olan elemandır. Aşağıdaki tabloda indeksleri ve değerleri bir arada görebilirsiniz.



## ➤ LİSTE ELEMANLARINA ERİŞİM

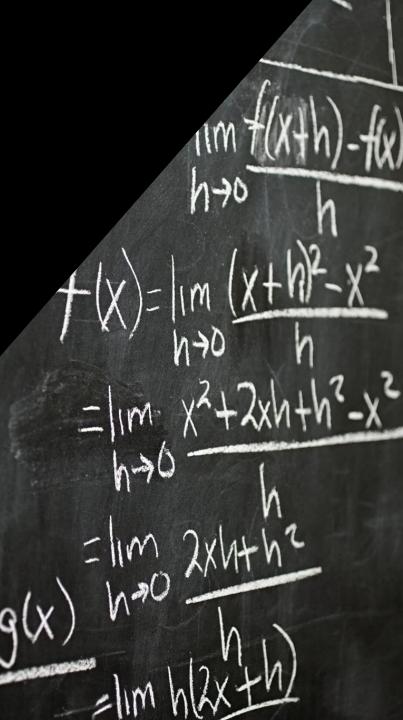
```
my_list = [10, 20, 30, 40, 50]

print(my_list[0]) # 10
print(my_list[2]) # 30
print(my_list[-1]) # 50 (negatif indeks sondan başlar)
```



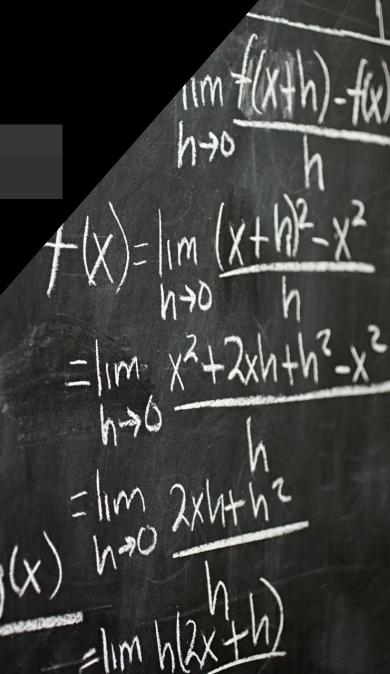
## > ÖRNEK

Renkler listesi oluşturarak indeksi 2 olan elemanı ekrana yazdırınız.



## > CEVAP

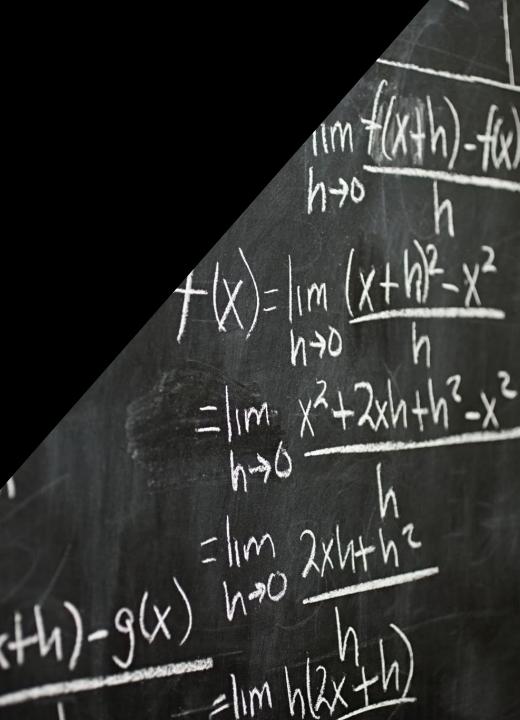
```
renkler=["Kırmızı", "Beyaz", "Sarı", "Yeşi", "Mavi"]
print(renkler[2])
```



Aşağıdaki kodun çıktısını yazınız (Python'da tek karakterden oluşan değerleri tek tırnak (') içinde tanımlayabilirsiniz.).

ders=['K','O','D','L','A','M','A']

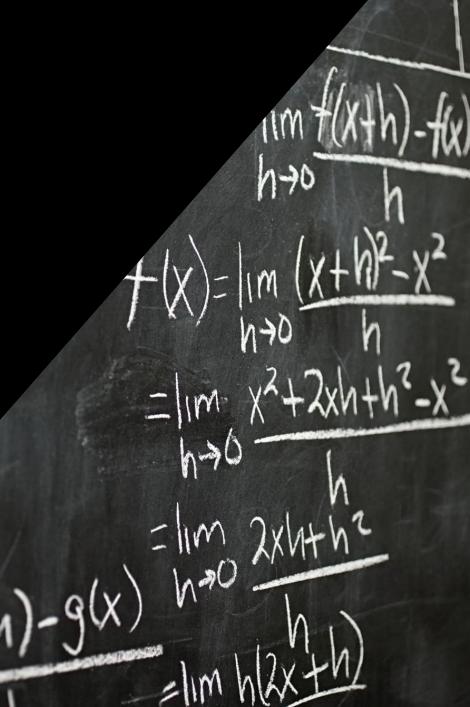
Kod satırı	Çıktı
print(ders[0])	
print(ders[2])	
print(ders[5])	



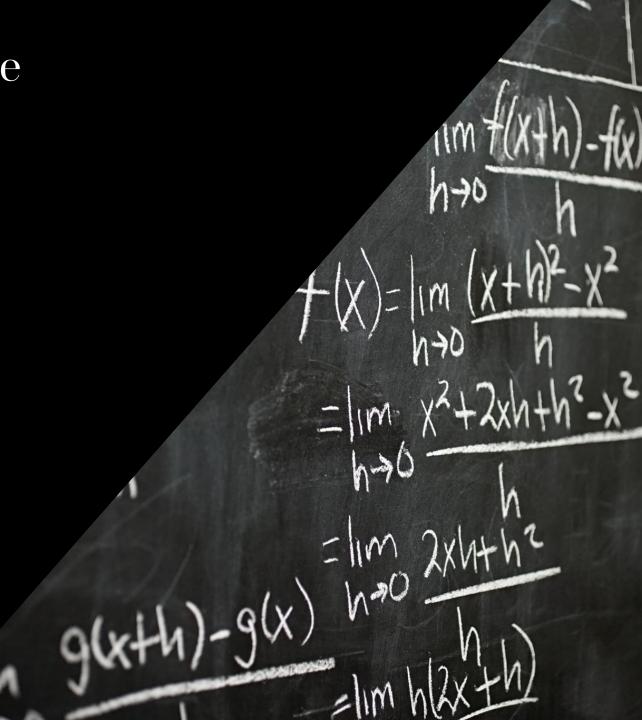
Indeksler negatif olarak da yazılabilir. Örneğin -l indeksi sondaki elemanı gösterirken -2 indeksi sondan bir öncekini gösterir.

ers=[K,O,D,L,A,M,A]	
Kod satırı	Çıktı
print (ders[-1])	
print(ders[-4])	
print(ders[-3])	

dere-FWYOYDYYLYAYAAYAA



Listelerde indekslerle birlikte iki nokta (:) operatörü kullanılarak istenilen elemanlara ulaşılabilir. Bu işlem için liste[başlangıç indeksi:bitiş indeksi] yapısı kullanılır.



#### Örnek :

```
asal_sayilar=[2, 3, 5, 7, 11, 13, 17, 19, 23]
print(asal_sayilar[1:4])
```

Bu örnekte indeksi 1 olan elemandan başlayarak indeksi 4 olan elemana (4 dâhil değil) kadar ekrana yazdırır. Dolayısıyla ekran çıktısı [3, 5, 7] olacaktır.

#### Örnek :

```
asal_sayilar=[2, 3, 5, 7, 11, 13, 17, 19, 23]
print(asal_sayilar[5:])
```

#### Ekran Çıktısı:

[13, 17, 19, 23]

Buradaki kullanımda dikkat edilirse başlangıç olarak 5 indeksi verilip bitiş indeksi ise verilmemiştir. Bu kullanımda indeksi 5 olan elemandan başlayarak son elemana kadar yazılır.

#### Örnek :

asal\_sayilar=[2, 3, 5, 7, 11, 13, 17, 19, 23] print(asal\_sayilar[:5])

#### Ekran Çıktısı:

[2, 3, 5, 7, 11]

Bu kullanımda da başlangıç indeksi verilmemiş bitiş indeksi olarak 5 verilmiştir. Başlangıç indeksinin verilmediği durumda indeksi 0 (sıfır) olan elemandan başlayarak yazdırılır.

#### Örnek:

asal\_sayilar=[2, 3, 5, 7, 11, 13, 17, 19, 23]

print(asal\_sayilar[0:6:2])

#### Ekran Çıktısı:

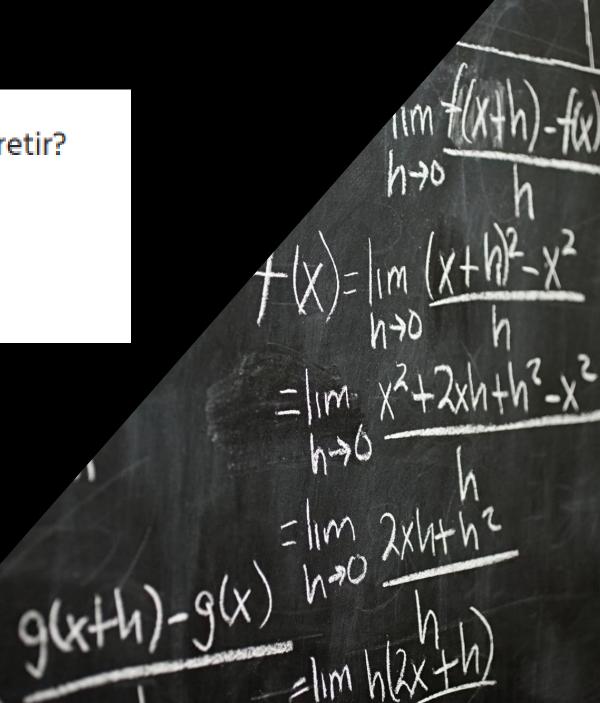
[2, 5, 11]

Bu kullanımda ise sırasıyla başlangıç indeksi, bitiş indeksi ve atlama değeri verilmiştir. Yani 0. indeksten başlayarak 6. indekse kadar ikişer artarak ekrana yazdırılır.

Sıra Sizde: Aşağıdaki kod nasıl bir çıktı üretir?

asal\_sayilar=[2, 3, 5, 7, 11, 13, 17, 19, 23]

print(asal\_sayilar[::2])



Aşağıdaki işlemlerin sonucunu yazınız.

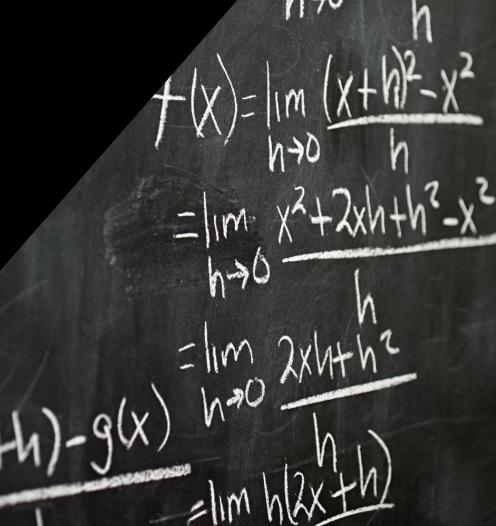
tek\_sayilar=[3, 5, 7, 9, 11, 13, 15, 17, 19]

print(tek_sayilar[0:6])	
print(tek_sayilar[2:5])	
print(tek_sayilar[3:8])	
print(tek_sayilar[:5])	
print(tek_sayilar[3:])	
print(tek_sayilar[0:8:2])	
print(tek_sayilar[::3])	

## > Liste Elemanını Değiştirme

Liste veri tipindeki bir elemanın indeksi kullanılarak yeni değer atanabilir.

```
my_list = [1, 2, 3, 4, 5]
my_list[2] = 10
print(my_list) # [1, 2, 10, 4, 5]
```



#### Örnek :

ornek=['Y','A','N','I','T'] #YANIT kelimesinin harflerinden bir liste oluşturuldu.

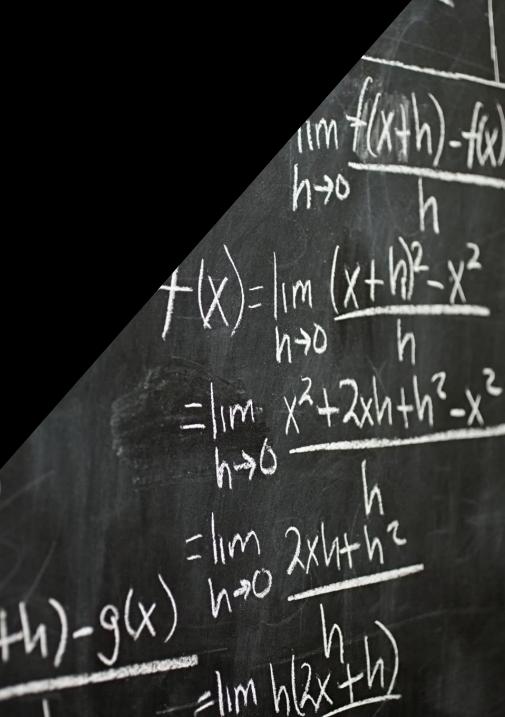
print(ornek) #Liste ekrana yazdırıldı.

ornek[0]='K' #Listenin ilk elemanı (indeksi sıfır) K olarak değiştirildi.

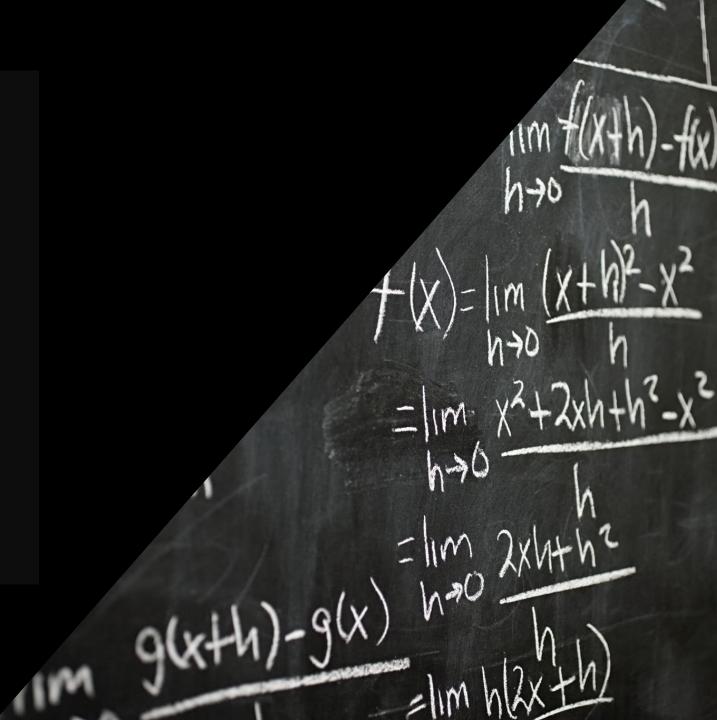
print(ornek) #Listenin yeni değeri KANIT kelimesinin harflerine dönüştü.

## SIRA SIZDE

Değerleri sırasıyla 3,1,2 olan listeyi 1,1,2 olarak değiştiriniz.

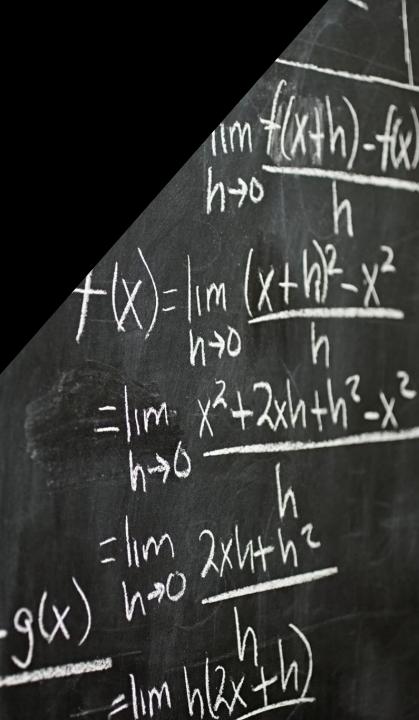


```
# Orijinal liste
liste2 = [3, 1, 2]
# Değiştirme işlemi
liste2[0] = 1
liste2[1] = 1
# Değiştirilmiş liste
print(liste2)
```

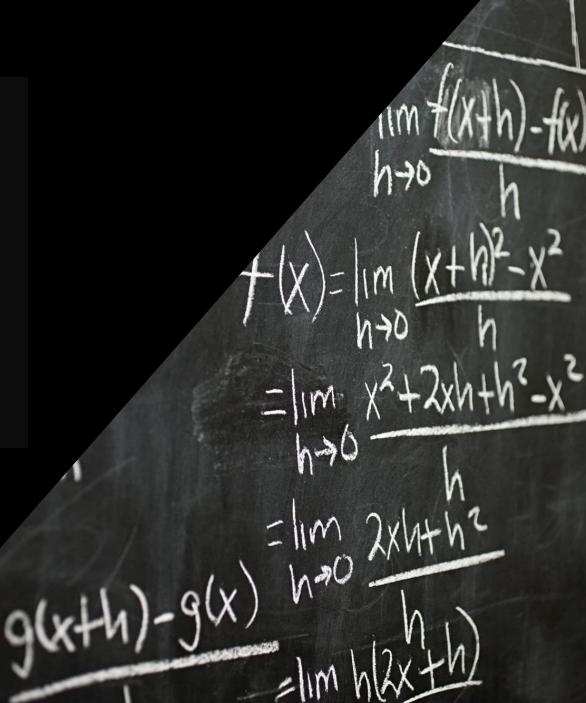


## SIRA SIZDE

Elemanları sırasıyla sanat, sanat, içindir olan listeyi sanat, toplum, içindir şeklinde değiştiriniz.



```
# Orijinal liste
liste1 = ["sanat", "sanat", "içindir"]
# Değiştirme işlemi
liste1[1] = "toplum"
# Değiştirilmiş liste
print(liste1)
```



#### **Liste Metodları:**

Python listeleri üzerinde birçok özel metod bulunur. Örneğin, eleman eklemek veya kaldırmak için kullanılabilirler.