

```
In [1]: # metin içindeki özel karakterlere göre (nokta, virgül, boşluk vb.)  
# metni list'e çevirme  
text="Ercan,Hakan,Melih,Osman"  
text
```

```
Out[1]: 'Ercan,Hakan,Melih,Osman'
```

```
In [2]: # split fonksiyonu metni istenilen karaktere göre ayırır  
word_list = text.split(',')  
word_list
```

```
Out[2]: ['Ercan', 'Hakan', 'Melih', 'Osman']
```

```
In [3]: # liste elemanlarını metine (str) çevirme  
# join fonk. liste içindeki elemanları biraraya getirir  
my_list=["a","b","c"]  
my_list
```

```
Out[3]: ['a', 'b', 'c']
```

```
In [4]: # listeyi birleştirir  
text = "".join(my_list)  
text
```

```
Out[4]: 'abc'
```

```
In [5]: # listeyi "-" karakteri ile birleştirir  
text = "-".join(my_list)  
text
```

```
Out[5]: 'a-b-c'
```

```
In [6]: # listeyi "," karakteri ile birleştirir  
text = ",".join(my_list)  
text
```

```
Out[6]: 'a,b,c'
```

```
In [10]: # metin içinden noktalama işaretlerini almadan  
# sadece kelimeleri alıp liste elde etme  
# re modülünü yükleme  
import re  
text="Merhaba, bugün, nasılsınız?"  
# re modülünde yer alan findall fonksiyonu kullanılır  
regulated_text=re.findall(r'[\w]+',text)  
regulated_text
```

```
Out[10]: ['Merhaba', 'bugün', 'nasılsınız']
```

```
In [11]: metin=input('İçerisinde noktalama işaretleri geçen bir metin giriniz:')  
İçerisinde noktalama işaretleri geçen bir metin giriniz>Hello world!
```

```
In [12]: duzenli_ifade=re.findall(r'[\w]+',metin)  
duzenli_ifade
```

```
Out[12]: ['Hello', 'world']
```

```
In [14]: import re  
def noktalama_isareti_at(metin):  
    return re.findall(r'[\w]+',metin)
```

```
In [15]: metin=input('İçerisinde noktalama işaretleri geçen bir metin giriniz:')  
duzenli_ifade=noktalama_isareti_at(metin)  
duzenli_ifade
```

```
İçerisinde noktalama işaretleri geçen bir metin giriniz:"fizik","kimya" ?? +*- ! #  
Out[15]: ['fizik', 'kimya']
```

```
In [16]: metin=input('İçerisinde noktalama işaretleri geçen bir metin giriniz:')  
duzenli_ifade=noktalama_isareti_at(metin)  
duzenli_ifade
```

```
İçerisinde noktalama işaretleri geçen bir metin giriniz:Şimdi en açık renginde gözlerin  
Out[16]: ['Şimdi', 'en', 'açık', 'renginde', 'gözlerin']
```

```
In [19]: # max fonk. ile liste içerisindeki en büyük sayısal değere sahip olan eleman bulunur  
numbers=[1,7,3]  
max_number = max(numbers)  
max_number
```

```
Out[19]: 7
```

```
In [20]: # min fonk. ile liste içerisindeki en küçük sayısal değere sahip olan eleman bulunur  
numbers=[1,7,3]  
min_number = min(numbers)  
min_number
```

```
Out[20]: 1
```

```
In [21]: # max fonk. string değerlerde alfabetik olarak en sondaki metni bulur  
courses=["fizik","yapay zeka","matematik"]  
max_course=max(courses)  
max_course
```

```
Out[21]: 'yapay zeka'
```

```
In [22]: # min fonk. string değerlerde alfabetik olarak en baştaki metni bulur  
courses=["fizik","yapay zeka","matematik"]  
min_course=min(courses)  
min_course
```

```
Out[22]: 'fizik'
```

```
In [23]: # extend fonk. ile listenin sonuna başka liste ekleme  
cities=['Ankara','Adana','Bursa']  
other_cities=['İzmir','İstanbul']  
cities.extend(other_cities)  
cities
```

```
Out[23]: ['Ankara', 'Adana', 'Bursa', 'İzmir', 'İstanbul']
```

```
In [24]: # extend fonk. alternatif olarak += operatörü kullanılabilir  
cities=['Ankara','Adana','Bursa']  
other_cities=['İzmir','İstanbul']  
cities += other_cities  
cities
```

```
Out[24]: ['Ankara', 'Adana', 'Bursa', 'İzmir', 'İstanbul']
```

```
In [25]: # iki listenin elemanını değiştirmeden üçüncü listede birleştirme  
cities=['Ankara','Adana','Bursa']  
other_cities=['İzmir','İstanbul']  
all_cities = cities + other_cities
```

```
In [26]: cities
```

```
Out[26]: ['Ankara', 'Adana', 'Bursa']
```

```
In [27]: other_cities
```

Out[27]: ['İzmir', 'İstanbul']

In [28]: all_cities

Out[28]: ['Ankara', 'Adana', 'Bursa', 'İzmir', 'İstanbul']

```
In [29]: # Liste içine başka bir liste ekleme
cities=['Ankara','Adana','Bursa']
other_cities=['İzmir','İstanbul']
# append listenin sonuna ekler
cities.append(other_cities)
cities
```

Out[29]: ['Ankara', 'Adana', 'Bursa', ['İzmir', 'İstanbul']]

```
In [31]: # cities listesinin tüm elemanları
for i in cities:
    print(i)
```

Ankara
Adana
Bursa
['İzmir', 'İstanbul']

```
In [32]: # cities listesinin eleman sayısı
len(cities)
```

Out[32]: 4

```
In [33]: # countries isimli boş liste oluşturma
countries=[]
# countries listesinin eleman sayısı
len(countries)
```

Out[33]: 0

```
In [34]: # append ile liste sonuna eleman eklenirken
# yeni listenin eleman sayısı 1 artar
numbers=[1,2,3]
numbers.append([4,5])
numbers
```

Out[34]: [1, 2, 3, [4, 5]]

```
In [35]: len(numbers)
```

Out[35]: 4

```
In [36]: # extend ile liste sonuna eleman eklenirken
# yeni listenin eleman sayısı kadar artar
numbers=[1,2,3]
numbers.extend([4,5])
numbers
```

Out[36]: [1, 2, 3, 4, 5]

```
In [37]: len(numbers)
```

Out[37]: 5

```
In [38]: # list içinde birden fazla aynı eleman varsa
# aynı olan elemanları tek elemene düşürme
numbers=[1,2,1,3,4,5,1,2,3]
numbers
```

Out[38]: [1, 2, 1, 3, 4, 5, 1, 2, 3]

```
In [39]: # numbers tipi list liste
type(numbers)
```

```
Out[39]: list
```

```
In [40]: # set ile liste kümeye dönüştürülür
# kümede bir eleman sadece 1 kez tanımlanır
numbers=set(numbers)
numbers
```

```
Out[40]: {1, 2, 3, 4, 5}
```

```
In [41]: # numbers tipi set (küme)
type(numbers)
```

```
Out[41]: set
```

```
In [42]: # list ile liste tipine dönüştürülür
numbers=list(numbers)
numbers
```

```
Out[42]: [1, 2, 3, 4, 5]
```

```
In [43]: # numbers tipi list (liste)
type(numbers)
```

```
Out[43]: list
```

```
In [44]: # yukarıdaki işlemleri iç içe tek satırda yapalım
numbers=[1,2,1,3,4,5,1,2,3]
numbers=list(set(numbers))
numbers
```

```
Out[44]: [1, 2, 3, 4, 5]
```

```
In [45]: # iki liste birbirine eşit mi?
numbers=[1,3,4,2]
values=[1,3,4,2]
# iki listenin eşitlik durumu == operatörü ile kontrol edilir
numbers == values
# elemanların liste içindeki sıraları eşit olduğundan True
```

```
Out[45]: True
```

```
In [46]: # iki liste birbirine eşit mi?
numbers=[1,3,4,2]
values=[1,3,2,4]
# iki listenin eşitlik durumu == operatörü ile kontrol edilir
numbers == values
# elemanların liste içindeki sıraları farklı olduğundan False
```

```
Out[46]: False
```

```
In [47]: # liste içerisindeki elemanları tek bir string olarak toplama
# ayraç olarak ',' kullanıldı
people = ['Hakan', 'Ercan', 'Melih', 'Osman']
text = ",".join(people)
text
```

```
Out[47]: 'Hakan,Ercan,Melih,Osman'
```

```
In [48]: def liste_elemanlarini_birlestir(liste,ayrac):
return ayrac.join(liste)
```

```
In [50]: liste = ['Hakan', 'Ercan', 'Melih', 'Osman']
ayrac=input('Liste birleştirmede kullanılacak ayraç karakterini giriniz:')
```

Liste birleştirmede kullanılacak ayraç karakterini giriniz:,

```
In [51]: liste_elemanlarini_birlestir(liste,ayrac)
```

```
Out[51]: 'Hakan,Ercan,Melih,Osman'
```

```
In [52]: ayrac=input('Liste birleştirmede kullanılacak ayraç karakterini giriniz:')
liste_elemanlarini_birlestir(liste,ayrac)
```

```
Out[52]: Liste birleştirmede kullanılacak ayraç karakterini giriniz:#
'Hakan#Ercan#Melih#Osman'
```

```
In [53]: # Liste içindeki tüm elemanları başka bir listede metin olarak (string) toplama
members=["Ercan",4,True,"Hakan",3]
members
```

```
Out[53]: ['Ercan', 4, True, 'Hakan', 3]
```

```
In [55]: for i in members:
print(i,type(i))
```

```
Ercan <class 'str'>
4 <class 'int'>
True <class 'bool'>
Hakan <class 'str'>
3 <class 'int'>
```

```
In [56]: str_members=[] # metin içerecek boş liste
for member in members: # tüm elemanlar
str_members.append(str(member)) # str ile metne dönüştürülür
str_members
```

```
Out[56]: ['Ercan', '4', 'True', 'Hakan', '3']
```

```
In [59]: # metinden ',' ile ayrılmış karakterleri listeye ayrı eleman olarak ekleme
# split() fonk. ile verilen metni ',' ayırılım
text = 'Hakan,Ercan,Melih,Osman'
values = text.split(',')
values
```

```
Out[59]: ['Hakan', 'Ercan', 'Melih', 'Osman']
```

```
In [60]: def metni_listeye_donustur(metin,ayrac):
return metin.split(ayrac)
```

```
In [61]: metin=input('Listeye dönüştürülecek metni giriniz:')
ayrac=input('Listeye dönüştürmede kullanılacak ayraç karakterini giriniz:')
metni_listeye_donustur(metin,ayrac)
```

```
Out[61]: Listeye dönüştürülecek metni giriniz:Ahmet,Mehmet,Furkan,Ayşe,Fatma
Listeye dönüştürmede kullanılacak ayraç karakterini giriniz:,
['Ahmet', 'Mehmet', 'Furkan', 'Ayşe', 'Fatma']
```

```
In [62]: values = (1,4,5)
values
```

```
Out[62]: (1, 4, 5)
```

```
In [63]: # tuple (demet) tipinde
type(values)
```

```
Out[63]: tuple
```

```
In [64]: # list() fonk. ile diğer tipteki değişkenler listeye dönüştürülür
values = list(values)
values
```

Out[64]: [1, 4, 5]

```
In [65]: # list (liste) tipinde  
         type(values)
```

Out[65]: list