

# Listeler

Python listelerinde liste elemanları sıralanabilir, güncellenebilir ayrıca her bir eleman liste içerisinde birden fazla tekrarlanabilir.

## 1.1. LİSTE OLUSTURMAK

Python'da bir liste oluşturmak için 3 yol vardır: [], list([iterable]) ve list comprehension

Python listeleri farklı değişken türlerini bir arada tutabilir.

```
liste = []
liste1 = [1, [2,3], (4,5), False, 'No']
liste2 = list(liste1)
liste3 = list([1,2,3])
print(type(liste1))
```

```
liste = [ifade for degisken in sequence]
liste = [x*2 for x in (1,2,3,4,5)] # [1, 4, 9, 16, 25]
squares = [i*i for i in range(10)] # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

## 1.2. LİSTE INDEXLEMEK VE PARÇALAMAK

Indexler 0'dan başlayıp eleman sayısından bir eksikliğine kadardır.

```
liste = ['a','b','c','d']
print(liste[2]) # 2. indexteki (baştan 3. eleman)
print(liste[:2]) # liste başından 2. indextekine kadar(2. index dahil değil)
print(liste[::-1]) # listeyi ters yazdırır
print(liste[:2])
# 0. indexten başlayarak 2 ve katları olan indexlileri yazdırır (slice stepping)
```

Eğer index olarak negatif bir sayı girerseniz, Python o sayıya listenin uzunluğunu ekler. 4 elemanlı bir listede -1. index aynı zamanda 3. indextir, -2. index aynı zamanda 2. indextir.

String parçalayarak liste oluşturabiliriz.

```
a = "Python Eğitim Serisi"
liste = list(a)
print(liste)
```

LİSTE	list =	[	'a',	'b',	'c',	'd' ]
INDEX		0	1	2	3	
NEGATİF INDEX		-4	-3	-2	-1	

## 1.3. LİSTEYE YENİ ELEMAN EKLEMEK VE ÇIKARMAK

Listeler değiştirilebilir, listeye bir eleman eklediğimizde yeni bir liste yaratılmaz.

```
list=["cookies", "pasta", "eggs"]
list += ["milk"] # [] gerekli
print(list)
a = b = c = d = list
# a b c d her biri aynı orijinal listeyi içeriyor fakat yeni liste yaratılmadı
a += ["bread"]
# a b c d'den herhangi birindeki listeyi değiştirirsek orijinal liste hepsinde değişir
print(list)
b.append("cream")
print(d) # print ile a/b/c/d/list hangisini yazdırırsak aynı çıktıyı verir
c.pop()
print(a)
a.insert(2,"spam")
print(b)
```

<code>liste_ismi.append(eklenecek_eleman)</code> methodu liste sonuna eleman ekler
<code>liste_ismi.insert(index, eklenecek_eleman)</code> methodu istenen indexe eleman ekler, ondan sonraki elemanların indexi 1 artırılır
<code>sabit_liste.extend(eklenecek_liste)</code> methodu bir listenin sonuna diğer bir listenin elemanlarını ekler, eklenecek_liste aynı kalır
<code>liste_ismi.pop(index)</code> methodu listeden o indexteki elemanı çıkarır ( <code>pop()</code> parantez boşsa sonuncuyu siler).( <code>pop</code> ayrıca silinen değeri geri döndürür)
<code>del liste_ismi[index]</code> fonksiyonu o indexteki elemanı siler (indexlemede öğrendiğimiz gibi <code>liste_ismi[1:4]</code> diyerek 1. indexten 4'e(dahil değil) kadar hepsini silebiliriz)
<code>liste_ismi.remove(silinecek_değer)</code> methodu istenen elemanı siler

#### 1.4. ELEMAN DEĞİSTİRMEK

```
liste = [1,2,3,4,5]
liste[3] = 100
print(liste)
liste[:2] = [10, 20]
print(liste)
liste[:3] = [0]
# liste başından 3.indexe kadar(dahil değil) olan elemanların yerine sadece 9 koydu ve
# listenin eleman sayısını (eşitliğin sağındaki eleman sayısı(bu durumda sadece 1)) -
# (yerine_koyduğu_eleman_sayısı(bu durumda 3)) kadar azalttı
print(liste)
liste*=2
print(liste)
```

#### 1.5. LİSTE BİRLEŞTİRMEK

```
liste1 = [1, 2, 3]
liste2 = [4, 5, 6]
yeniliste = liste1 + liste2
print(yeniliste)
print(liste1)
print(liste2)
liste1.extend(liste2)
print(liste1)
print(liste2)
```

#### 1.6. İÇ İÇE LİSTE OLUSTURMAK

```
liste1 = [1, 2, 3]
liste2 = [4, 5, 6]
yeniliste = [liste1, liste2] # [[1, 2, 3], [4, 5, 6]]
print(yeniliste[1][2])      # 6
```

<code>len(liste_ismi)</code> fonksiyonu listenin eleman sayısını verir
<code>min(liste_ismi)</code> fonksiyonu listenin en küçük elemanını verir
<code>max(liste_ismi)</code> fonksiyonu listenin en büyük elemanını verir
<code>liste_ismi.count(eleman)</code> fonksiyonu listede o eleman kaç tane var onu döndürür
<code>liste_ismi.sort()</code> methodu listeyi küçükten büyüğe sıralar, orijinal listeyi değiştirir
<code>sorted(liste_ismi)</code> fonksiyonu kopya bir liste yaratıp onu küçükten büyüğe sıralar, orijinal liste değişmez
<code>liste_ismi.reverse()</code> methodu listeyi ters çevirir, orijinal liste değişir

enumerate() fonksiyonu ile aynı anda index ve o indexteki elemanı yazdırmak	
liste = ['a', 'b', 'c']	0 a    # her elemanı indexiyle bastırır
for index,eleman in enumerate(liste):	1 b
print(index,eleman)	2 c
len() fonksiyonu ile indexleri yazdırmak	
liste = ['a', 'b', 'c']	0        # sadece indexleri yazdırır
for index in range(len(liste)):	1
print(index)	2

## EXTRA KAYNAKLAR:

## (EN) PYTHON LISTS AND LIST MANIPULATION:

1. <https://towardsdatascience.com/python-basics-6-lists-and-list-manipulation-a56be62b1f95>
2. [https://www.tutorialspoint.com/python/python\\_lists.htm](https://www.tutorialspoint.com/python/python_lists.htm)

## (FR) LES LISTES:

1. <https://python.doctor/page-apprendre-listes-list-tableaux-tableaux-liste-array-python-cours-debutant>
1. <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/232026-creez-des-listes-et-des-tuples-1-2>

## (EN) LIST COMPREHENSION:

1. <https://realpython.com/list-comprehension-python/#when-not-to-use-a-list-comprehension-in-python>

## (EN) DATA STRUCTURES AND FUNCTIONS:

1. <https://docs.python.org/3/tutorial/datastructures.html>
2. <https://docs.python.org/3/library/functions.html>

## CHALLENGE 1

```
menu = [
    ["egg", "bacon"],
    ["egg", "sausage", "bacon"],
    ["egg", "spam"],
    ["egg", "bacon", "spam"],
    ["egg", "bacon", "sausage", "spam"],
    ["spam", "bacon", "sausage", "spam"],
    ["spam", "sausage", "spam", "bacon", "spam", "tomato", "spam"],
    ["spam", "egg", "spam", "spam", "bacon", "spam"],
]
```

Menudeki tüm spam'leri döngüler ve burada gösterilen fonksiyon ya da methodlarla menüden çıkartın ve menüyü yazdırın.

(Tip: iç içe for döngüsü kullanılarak her elemana erişilebilir)

## PROGRAM ÇIKTISI

Original Menu:

```
['egg', 'bacon']
['egg', 'sausage', 'bacon']
['egg', 'spam']
['egg', 'bacon', 'spam']
['egg', 'bacon', 'sausage', 'spam']
['spam', 'bacon', 'sausage', 'spam']
['spam', 'sausage', 'spam', 'bacon', 'spam', 'tomato', 'spam']
['spam', 'egg', 'spam', 'spam', 'bacon', 'spam']
```

Güncellenmiş Menu:

```
['egg', 'bacon']
['egg', 'sausage', 'bacon']
['egg']
['egg', 'bacon']
['egg', 'bacon', 'sausage']
['bacon', 'sausage']
['sausage', 'bacon', 'tomato']
['egg', 'bacon']
```

## CHALLENGE 2

Please add options from the list below

```
1: computer
2: monitor
3: keyboard
4: mouse
5: mouse mat
6: hdmi cable
0: to finish
```

Donguler ve fonksiyonlar ya da methodları kullanarak kullanıcının girdiği sayı değeri eğer üstteki menüde görülenlerdence "Adding input\_girilen\_sayi" desin ve menüde o sayının sağındaki parçayı listeye eklesin, eğer girilen sayı değeri menüde yoksa üstteki menüyü tekrar oynatsın, eğer girilen sayı 0'sa listeyi oynatsın ve program sonlansın.  
(Tip: girilen sayıları int demeden input() ile alırsanız bir sayı ile denkliğini if içinde böyle bulabilirsiniz: current\_choice == '1')

## PROGRAM ÇIKTISI

Please add options from the list below

```
1: computer
2: monitor
3: keyboard
4: mouse
5: mouse mat
6: hdmi cable
0: to finish
```

1

Adding 1

2

Adding 2

7

Please add options from the list below

```
1: computer
2: monitor
3: keyboard
4: mouse
5: mouse mat
6: hdmi cable
0: to finish
```

5

Adding 5

0

```
['computer', 'monitor', 'mouse mat']
```