In [1]:
```
# içinde elemanı olmayan boş liste tanımlama
bos_liste=[]
```

In [2]:
```
bos_liste
```

Out[2]:
```
[]
```

In [3]:
```
# tek elemanlı bir liste
liste=['Python']
liste
```

Out[3]:
```
['Python']
```

In [4]:
```
# birden fazla elemanlı liste
liste=['Python','C#','Java','PHP', 3 , 5, 12, 3.14]
liste
```

Out[4]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14]
```

In [5]:
```
# listede bulunan elemanlar değiştirilebilir türdedir.
# liste değiştirilebilir veri tipidir.
```

In [6]:
```
# liste isimli liste değişkeninin kimlik numarası
id(liste)
```

Out[6]:
```
2560474457664
```

In [7]:
```
# bos_liste isimli liste değişkeninin kimlik numarası
id(bos_liste)
```

Out[7]:
```
2560474459456
```

In [8]:
```
# liste elemanlarının değerini değiştirme
liste=[1,2,3,4,5,6,7]
```

In [9]:
```
liste
```

Out[9]:
```
[1, 2, 3, 4, 5, 6, 7]
```

In [10]:
```
# elemanlarının değeri değiştirilen listenin kimlik numarası da değişir
# Bu durum liste değerlerinin farklı hafıza bölgesinde tutulmasından kaynaklanmaktadır
id(liste)
```

Out[10]:
```
2560474464512
```

In [11]:
```
liste
```

Out[11]:
```
[1, 2, 3, 4, 5, 6, 7]
```

In [12]:
```
# append listenin sonuna eleman ekler
liste.append('Python')
liste
```

Out[12]:
```
[1, 2, 3, 4, 5, 6, 7, 'Python']
```

In [13]:
```
# sonuna eleman eklenince kimlik değeri değişmedi
# listeler değiştirilebilir veri tipleridir
id(liste)
```

Out[13]:
```
2560474464512
```

In [14]:
```
# list() ile boş liste oluşturulur
bos_liste=list()
```

```
bos_liste
```

Out[14]:
```
[]
```

In [15]:
```
# list() fonksiyonuna karakter dizisi verilirse
# karakter dizisinin her bir elemanı listenin bir elemanı olur
liste=list('Merhaba')
liste
```

Out[15]:
```
['M', 'e', 'r', 'h', 'a', 'b', 'a']
```

In [16]:
```
# # liste_olustur() fonksiyonunu tanımlama
def liste_olustur(metin):
    liste=list(metin)
    return liste
```

In [17]:
```
# liste_olustur() fonksiyonunu deneme
liste_olustur('python')
```

Out[17]:
```
['p', 'y', 't', 'h', 'o', 'n']
```

In [18]:
```
# klavyeden metin okuma
metin=input('Lütfen listeye eleman eklenecek metni giriniz:')
```

```
Lütfen listeye eleman eklenecek metni giriniz:birecik myo
```

In [19]:
```
# tanımlanan fonksiyonu çağırma
liste_olustur(metin)
```

Out[19]:
```
['b', 'i', 'r', 'e', 'c', 'i', 'k', ' ', 'm', 'y', 'o']
```

In [20]:
```
# list() fonksiyonuna sayı değeri verilmez
# list() fonksiyonuna tamsayı tipli değer eklenemez
# list()
liste=list(12)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [20], in <cell line: 4>()
      1 # list() fonksiyonuna sayı değeri verilmez
      2 # list() fonksiyonuna tamsayı tipli değer eklenemez
      3 # list()
----> 4 liste=list(12)

TypeError: 'int' object is not iterable
```

In [21]:
```
# TypeError: 'int' object is not iterable
# Tip Hatası: 'int' nesnesi yinelenemez
```

In [22]:
```
# list() fonksiyonuna sayı değeri verilmez
# list() fonksiyonuna ondalıklı sayı tipli değer eklenemez
liste=list(12.25)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [22], in <cell line: 3>()
      1 # list() fonksiyonuna sayı değeri verilmez
      2 # list() fonksiyonuna ondalıklı sayı tipli değer eklenemez
----> 3 liste=list(12.25)

TypeError: 'float' object is not iterable
```

In [23]:
```
# tuple(demet) veri yapısı
demet=(1,2,3,4)
demet
```

Out[23]:
```
(1, 2, 3, 4)
```

In [24]:
```
type(demet)
```

Out[24]:
```
tuple
```

In [25]:
```
# tuple veri tipini list fonksiyonuna gönderme
liste=list(demet)
liste
```

Out[25]:
```
[1, 2, 3, 4]
```

In [26]:
```
type(liste)
```

Out[26]:
```
list
```

In [27]:
```
for i in range(10):
    print(i, end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

In [28]:
```
# range() fonksiyonu ile üretilen değerleri listeye dönüştürebiliriz
liste=list(range(10))
liste
```

Out[28]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [29]:
```
liste2=[0,1,2,3,4,5,6,7,8,9]
liste2
```

Out[29]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [30]:
```
# listeden range fonksiyonuna dönüşüm olmaz
x=range(liste2)
x
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [30], in <cell line: 2>()
      1 # listeden range fonksiyonuna dönüşüm olmaz
----> 2 x=range(liste2)
      3 x

TypeError: 'list' object cannot be interpreted as an integer
```

In [31]:
```
liste
```

Out[31]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [32]:
```
# listenin ilk elemanına erişim
liste[0]
```

Out[32]:
```
0
```

In [33]:
```
liste=['Python','C#','Java','PHP', 3 , 5, 12, 3.14]
liste
```

Out[33]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14]
```

In [34]:
```
# listenin ikinci elamanına erişim
liste[1]
```

Out[34]:
```
'C#'
```

In [35]:
```
# listenin son elamanına erişim
liste[-1]
```

Out[35]:
```
3.14
```

In [36]:
```
# listenin sondan ikinci elamanına erişim
liste[-2]
```

Out[36]:
```
12
```

In [37]:
```
liste2
```

Out[37]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [38]:
```
# listenin sonuna liste2 listesini ekler
liste += liste2
```

In [39]:
```
liste
```

Out[39]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [40]:
```
# listenin sonuna tamsayı eleman ekleme
liste=['Python','C#','Java','PHP', 3 , 5, 12, 3.14]
liste += [10]
liste
```

Out[40]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10]
```

In [41]:
```
# listenin sonuna string tipinde eleman ekleme
liste += ['10']
liste
```

Out[41]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10, '10']
```

In [42]:
```
liste
```

Out[42]:
```
['Python', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10, '10']
```

In [43]:
```
# insert() fonk. ile listede araya eleman eklenir
# insert() fonk. ilk parametresi eklenecek indeks numarası
# insert() fonk. ikinci parametresi eklenecek eleman
# 1. indekse (2. elemana) 'C' değeri eklenir.
# Araya eklendiğinden diğer indeksteki elemanlar kaydırılır.
liste.insert(1,'C')
liste
```

Out[43]:
```
['Python', 'C', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10, '10']
```

In [44]:
```
# 2. indekse (3. elemana) 'C++' değeri eklenir.
liste.insert(2,'C++')
liste
```

Out[44]:
```
['Python', 'C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10, '10']
```

In [45]:
```
# listeden eleman çıkarma için -= kullanılamadı
liste -= 'Java'
liste
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [45], in <cell line: 2>()
      1 # listeden eleman çıkarma için -= kullanılamadı
----> 2 liste -= 'Java'
      3 liste

TypeError: unsupported operand type(s) for -=: 'list' and 'str'
```

In [46]:
```
# remove() listeden eleman çıkarma için kullanılır
liste.remove('10')
liste
```

Out[46]:
```
['Python', 'C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 12, 3.14, 10]
```

In [47]:
```
# liste fonksiyonu tek parametre alır
# listeden birden fazla eleman silemedi
liste.remove(12,3.14)
liste
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [47], in <cell line: 3>()
      1 # liste fonksiyonu tek parametre alır
      2 # listeden birden fazla eleman silemedi
----> 3 liste.remove(12,3.14)
      4 liste

TypeError: list.remove() takes exactly one argument (2 given)
```

In [48]:
```
liste.remove(3.14)
liste
```

Out[48]:
```
['Python', 'C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 12, 10]
```

In [49]:
```
liste.remove(12)
liste
```

Out[49]:
```
['Python', 'C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 10]
```

In [50]:
```
# listede olmayan eleman kaldırılamaz
liste.remove(4)
liste
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [50], in <cell line: 2>()
      1 # listede olmayan eleman kaldırılamaz
----> 2 liste.remove(4)
      3 liste

ValueError: list.remove(x): x not in list
```

In [51]:
```
liste
```

Out[51]:
```
['Python', 'C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 10]
```

In [52]:
```
# listeden ilk elemanı silme
liste.remove(liste[0])
liste
```

Out[52]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3, 5, 10]
```

In [53]:
```
# listeden son elemanı silme
liste.remove(liste[-1])
liste
```

Out[53]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3, 5]
```

In [54]:
```
# listenin eleman sayısı
len(liste)
```

Out[54]:
```
7
```

In [55]:
```
# listeden son elemanı silme
liste.remove(liste[-1])
liste
```

Out[55]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3]
```

In [56]:
```
# listenin eleman sayısı
len(liste)
```

Out[56]:
```
6
```

In [57]:
```
# listelerin metodları
for i in dir(list()):
    if '_' not in i:
        print(i)
```

```
append
clear
copy
count
extend
index
insert
pop
remove
reverse
sort
```

In [58]:
```
# listede 3 elamanı kaç tane var
liste.count(3)
```

Out[58]:
```
1
```

In [59]:
```
liste.append(3)
liste
```

Out[59]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3, 3]
```

In [60]:
```
# listede 3 elamanı kaç tane var
liste.count(3)
```

Out[60]:
```
2
```

In [61]:
```
# listede 5 elamanı kaç tane var
liste.count(5)
```

Out[61]:
```
0
```

In [62]:
```
# ilgili elemanın indeksini bulur
liste.index('Java')
```

Out[62]:
```
3
```

In [63]:
```
# ilgili elemanın indeksini bulur
liste.index('C')
```

Out[63]:
```
0
```

In [64]:
```
# ilgili elemanın indeksini bulur
# ilk bulunan konumun indeks değerini alır
liste.index(3)
```

Out[64]:
```
5
```

In [65]:
```
liste
```

Out[65]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3, 3]
```

In [66]:
```
# pop() fonk. son elemanı kaldırır
liste.pop()
liste
```

Out[66]:
```
['C', 'C++', 'C#', 'Java', 'PHP', 3]
```

In [67]:
```
# pop() fonk. son elemanı kaldırır
liste.pop()
liste
```

Out[67]:
```
['C', 'C++', 'C#', 'Java', 'PHP']
```

In [68]:
```
# listenin çift indeksli değerlerini yazdırma
for i in liste:
    indeks = liste.index(i)
    if indeks % 2 == 0:
        print(i, end=' ')
```

```
C C# PHP
```

In [69]:
```
# listenin tek indeksli değerlerini yazdırma
for i in liste:
    indeks = liste.index(i)
    if indeks % 2 == 1:
        print(i, end=' ')
```

```
C++ Java
```

In [70]:
```
satranc=['Kale','At','Fil','Şah','Fil','At','Kale']
satranc
```

Out[70]:
```
['Kale', 'At', 'Fil', 'Şah', 'Fil', 'At', 'Kale']
```

In [71]:
```
renk=input('Taş rengini giriniz (S/B) :')
renk=renk.lower() # küçük harfe çevir
renk
```

```
Taş rengini giriniz (S/B) :B
```

Out[71]:
```
'b'
```

In [72]:
```
sah_indeks=satranc.index('Şah')
sah_indeks
```

Out[72]:
```
3
```

In [73]:
```
# renk siyah ise vezir şah konumundan sonraya eklenir
if renk == 's':
    satranc.insert(sah_indeks+1,'Vezir')
    print(satranc)
elif renk == 'b':
    satranc.insert(sah_indeks,'Vezir')
    print(satranc)
else:
    print('Hatalı renk değeri!')
```

```
['Kale', 'At', 'Fil', 'Vezir', 'Şah', 'Fil', 'At', 'Kale']
```

In [74]:
```
satranc=['Kale','At','Fil','Şah','Fil','At','Kale']
satranc
```

Out[74]:
```
['Kale', 'At', 'Fil', 'Şah', 'Fil', 'At', 'Kale']
```

In [75]:
```
renk=input('Taş rengini giriniz (S/B) :')
renk=renk.lower() # küçük harfe çevir
renk
```

```
Taş rengini giriniz (S/B) :S
```

Out[75]:
```
's'
```

In [76]:
```
# renk siyah ise vezir şah konumundan sonraya eklenir
if renk == 's':
    satranc.insert(sah_indeks+1,'Vezir')
```

```
    print(satranc)
elif renk == 'b':
    satranc.insert(sah_indeks,'Vezir')
    print(satranc)
else:
    print('Hatalı renk değeri!')
```

```
['Kale', 'At', 'Fil', 'Şah', 'Vezir', 'Fil', 'At', 'Kale']
```

In [79]:
```
def satranc_vezir_ekle(renk,tas):
    if tas == 'Vezir':
        # renk siyah ise vezir şah konumundan sonraya eklenir
        if renk == 's':
            satranc.insert(sah_indeks+1,tas)
            print(satranc)
        # renk beyaz ise vezir şah konumundan önceye eklenir
        elif renk == 'b':
            satranc.insert(sah_indeks,tas)
            print(satranc)
        else:
            print('Hatalı renk değeri!')
    else:
        print('Vezir dışında bir taş girildi!')
```

In [80]:
```
satranc=['Kale','At','Fil','Şah','Fil','At','Kale']
satranc
```

Out[80]:
```
['Kale', 'At', 'Fil', 'Şah', 'Fil', 'At', 'Kale']
```

In [81]:
```
satranc_vezir_ekle('s','vezir')
```

```
Vezir dışında bir taş girildi!
```

In [82]:
```
satranc_vezir_ekle('s','Vezir')
```

```
['Kale', 'At', 'Fil', 'Şah', 'Vezir', 'Fil', 'At', 'Kale']
```

In [83]:
```
renk=input('Taş rengini giriniz (S/B) :')
renk=renk.lower() # küçük harfe çevir
renk
```

```
Taş rengini giriniz (S/B) :S
```

Out[83]:
```
's'
```

In [84]:
```
tas=input('Taşı giriniz (Vezir) :')
tas=tas.lower() # küçük harfe çevir
tas
```

```
Taşı giriniz (Vezir) :Vezir
```

Out[84]:
```
'vezir'
```

In [85]:
```
satranc=['Kale','At','Fil','Şah','Fil','At','Kale']
satranc
```

Out[85]:
```
['Kale', 'At', 'Fil', 'Şah', 'Fil', 'At', 'Kale']
```

In [86]:
```
satranc_vezir_ekle(renk,tas)
```

```
Vezir dışında bir taş girildi!
```

In [87]:
```
tas='Vezir'
tas
```

Out[87]:
```
'Vezir'
```

In [88]:
```
renk=input('Taş rengini giriniz (S/B) :')
renk=renk.lower() # küçük harfe çevir
renk
```

```
Taş rengini giriniz (S/B) :S
```

Out[88]:
```
's'
```

In [90]:
```
def satranc_vezir_ekle(renk,tas):
    satranc=['Kale','At','Fil','Şah','Fil','At','Kale']
    # renk siyah ise vezir şah konumundan sonraya eklenir
    if renk == 's':
        satranc.insert(sah_indeks+1,tas)
        print(satranc)
    # renk beyaz ise vezir şah konumundan önceye eklenir
    elif renk == 'b':
        satranc.insert(sah_indeks,tas)
        print(satranc)
    else:
        print('Hatalı renk değeri!')
```

In [91]:
```
satranc_vezir_ekle(renk,tas)
```

```
['Kale', 'At', 'Fil', 'Şah', 'Vezir', 'Fil', 'At', 'Kale']
```

In [92]:
```
renk=input('Taş rengini giriniz (S/B) :')
renk=renk.lower() # küçük harfe çevir
renk
```

```
Taş rengini giriniz (S/B) :B
```

Out[92]:
```
'b'
```

In [93]:
```
satranc_vezir_ekle(renk,tas)
```

```
['Kale', 'At', 'Fil', 'Vezir', 'Şah', 'Fil', 'At', 'Kale']
```