

Veri Yapıları

List:

*Köşeli parantez içerisinde aralarına virgül koyularak sıralanır ör: L1=[1,2,3,4]

*Listenin içindeki elemanlar numaralandırılır buna indis numarası denilir bu sistem 0 dan başlayarak yapılır.

*Bir elemana erişmek için Listede[indisnumarası] kullanılır ör: L=[1,2,3,4,5]

L[0]=1

L[1]=2

Liste metotları:

Metot adı	Açıklama	Örnek
L.append(x)	Liste sonuna x elemanını ekler.	L=[2,3,4] L.append(10) print(L) # [2,3,4,10]
L.extend(L2)	Liste sonuna L2 listesini ekler.	L1=[2,3,4] L2=[5,6] L1.extend(L2) print(L1) # [2,3,4,5,6]
L.insert(i,x)	x elemanını L listesinin i. indisli eleman olarak ekler.	L=[2,4,6,8] L.insert(2,15) print(L) # [2,4,15,6,8]
L.remove(x)	Listedeki ilk x değerini siler.	L=[2,4,6,8,4] L.remove(4) print(L) # [2,6,8,4]

Metot adı	Açıklama	Örnek
L.copy()	Listenin bir kopyasını oluşturur.	L1=[2,4,6,8] L2=L1.copy() print(L2) # [2,4,6,8]
L.reverse()	Liste elemanlarını tersten oluşturur.	L=[2,4,6] L.reverse() print(L) # [6,4,2]
L.sort()	Aynı tip elemanlardan oluşan listeyi küçükten büyüğe sıralar.	L=[8,7,1,3,9,15,2] L.sort() print(L) # [1,2,3,7,8,9,15]
len(liste)	Listedeki eleman sayısını verir.	L=[4,5,8,1,2,4] adet=len(L) print(adet) # 6

Metot adı	Açıklama	Örnek
L.pop(i)	Listeden i indis numaralı elemanı siler.	L=[2,4,6,8] L.pop(2) print(L) # [2, 4, 8]
L.clear()	Listedeki tüm elemanları siler.	L=[2,4,6,8] L.clear() print(L) # []
L.index(x)	x elemanının indis numarasını verir.	L=[2,4,6,8] print(L.index(6)) # 2
L.count(x)	Listedeki x ile aynı olan elemanların sayısını verir.	L=[2,4,6,8,4,4] n=L.count(4) print(n) # 3

Kullanım için örnek:

Örnek 1: Verilen N tane sayıyı listeye **ekleyen**, sayıların **ortalamasını** bulan, en küçük sayıyı listeden **çıkaran** ve bu listeye göre sayıları **sıralayan** Python programını yazınız.

```

N=int(input('Sayı adedini giriniz:'))
L=[] #Boş liste
#Sayıları girme ve Listeye ekleme
for i in range(0,N):
    x=int(input('x='))
    L.append(x)
print(L)
#Ortalama bulma
top=0
for i in range(0,N):
    top=top+L[i]
ort=top/N
print(ort)
#En küçük eleman ve listeden çıkarma
ek=999999
for i in range(0,N):
    if L[i]<ek:
        ek=L[i]
    
```

print(ek)
ekindis=L.index(ek)
L.pop(ekindis)
print(L)
#En son listeyi sıralama
L.sort()
print(L)

L.remove(ek)
komutuda kullanılabilir.

Ekran Çıktısı:

```

Sayı adedini giriniz:6
x=7
x=4
x=2
x=9
x=5
x=8
[7, 4, 2, 9, 5, 8]
5.833333333333333
2
[7, 4, 9, 5, 8]
[4, 5, 7, 8, 9]
    
```

Listenin toplamı için: top=sum(L) de kullanılabilir.

Listeden en küçük bulma komutu için: ek=min(L) de kullanılabilir.

Çok Boyutlu Liste Matris: işlem boyutu kadar köşeli parantez eklenir

Örnek: İki matrisi toplayan Python programı

```

A=[[1,3,-1],[3,-2,5],[4,1,2]]
B=[[4,1,1],[-2,2,3],[5,0,1]]
C=[[0,0,0],[0,0,0],[0,0,0]]
for i in range(0,3):
    for j in range(0,3):
        C[i][j]=A[i][j]+B[i][j]
for k in C:
    print(k)
    
```

C matrisinin matris formunda çıktısı

Ekran Çıktısı:

```

[5, 4, 0]
[1, 0, 8]
[9, 1, 3]
    
```

$$A = \begin{bmatrix} 1 & 3 & -1 \\ 3 & -2 & 5 \\ 4 & 1 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} 4 & 1 & 1 \\ -2 & 2 & 3 \\ 5 & 0 & 1 \end{bmatrix}$$

İç içe döngüde indislerin değişimi:

i	j
0	0 → A ₀₀
	1 → A ₀₁
	2 → A ₀₂
1	0 → A ₁₀
	1 → A ₁₁
	2 → A ₁₂
2	0 → A ₂₀
	1 → A ₂₁
	2 → A ₂₂

Not: C matrisini yazdırmak için print(C)'de kullanılabilir. Bu durumda ekran çıktısı:

```
[[5, 4, 0], [1, 0, 8], [9, 1, 3]]
```

Liste Üreteçleri: amaç tek satırlık kodla liste elemanlarını oluşturmak

Örnek: 0'dan 100'e kadar sayılardan oluşan liste oluşturma

Klasik yol:

```
L=[]
for i in range(0,101):
    L=L+[i]
```

Liste üretici ile:

```
L=[i for i in range(0,101)]
```

Örnekler:

1. 100'den 5'e, beşer beşer azalan sayılardan oluşan listeyi oluşturma

```
L=[i for i in range(100,0,-5)]
```

2. 1'den 10'a kadar sayıların karelerinin 2'ye bölümünden oluşan listeyi oluşturma

```
L=[(x*x)/2 for x in range(1,11)]
```

Liste Dilimlenmesi:

Listenin Dilimlenmesi: Liste içinden veriler belli yerlerden kesilerek yeni liste oluşturulabilir.

```
listeadı[start:stop:step]
```

start: başlangıç indis no (dahil)

stop : bitiş indis no (dahil değil)

step : artım (kullanılmazsa 1 alınır)

Örnek: L=[2,4,6,8,10,12,14] listesi içinden indis numarası 3 ile 5 arasındaki verilerden yeni bir liste oluşturulması

```
L=[2,4,6,8,10,12,14]
yeniliste=L[3:5]
print(yeniliste)
```

Ekran Çıktısı:

```
[8, 10]
```

Fonksiyonlar:

parametrelili fonksiyon dışarıdan değer alır

parametresiz fonksiyon dışarıdan değer almayan

ana programa değer aktaran döndüren

ana programa değer aktarmayan döndürmeyen denir

fonksiyon içi doğru tablı yazılır ve başlangıcı şu şekilde olur

def fonksiyonadı():

komut1

komut2

```
def kare(x):
```

```
    sonuc=x*x
```

```
    return sonuc
```

Fonksiyon tanımı (Parametrelili, değer aktaran)

Ana programa değer döndürme

```
x=int(input('Karesi alınacak sayı='))
```

```
skare=kare(x)
```

Fonksiyon çağırma ve girilen x'in fonksiyona gönderilmesi

```
print('Sonuç=', skare)
```

```
print('10 un karesi=', kare(10))
```

```
print('12 nin karesi=', kare(12))
```

fonksiyonun içindeki return komutu bir değişkenin sonucunu ana programa gönderir

Dosyalar: Verilerin kaydedilmesi ve daha sonra kullanımı için varlardır ikiye ayrılır metin ve ikili dosyalar olarak

dosya açmak için open(dosya adı)

dosya kapatmak için close(dosya adı)

Dosya açma modları (metin dosyaları):

r	Sadece okumak için açılır. Dosyanın açılabilmesi için önceden oluşturulmuş olması gerekir. Bu modda açılmış olan bir dosyaya yazma yapılamaz.
w	Sadece yazmak için açılır. Dosya diskte olsun veya olmasın dosya yeniden oluşturulur. Aynı isimle dosya varsa var olanın içeriği silinir. Bu modda açılmış olan bir dosyadan veri okuma yapılamaz.
a	Dosyadan hem veri okuma, hem de veri yazma amaçlı açılır. Mevcut bir dosyanın sonuna veri eklenebilir.
x	w ile aynı işleve sahiptir. Farkı, eğer aynı isimle önceden dosya varsa silmez, hata mesajı verir.
r+	Okuma ve yazma. Dosyanın daha önceden var olması gerekir.
w+	Okuma ve yazma. Dosya önceden var ise dosya içi silinir, yok ise ilk defa oluşturulur.
a+	Okuma ve yazma. O isimde dosya yok ise ilk defa oluşturulur.
x+	Okuma ve yazma. Aynı isimle başka bir dosya var ise o dosya silinmez, hata mesajı verir.

Dosya açma modları (ikili dosyalar): Yukarıda verilen modlara b harfi eklenir: rb, wb, ab, xb, rb+, wb+, ab+, xb+

Dosya Komutları:

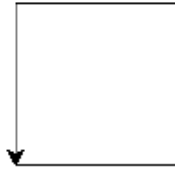
Komut	Görevi
open()	Dosya oluşturur, açar.
close()	Dosyayı kapatır.
write()	Dosyaya istenen veriyi yazar. String tipinde tek parametre alır.
writelines()	Dosyaya verilen listeyi (birden fazla satırlık veri) yazar.
read()	Tüm dosyayı okur. Sonucu string olarak döndürür.
read(n)	Dosyadan n karakterlik veri okur.
readline()	Dosyadan satır satır veri okur.
readlines()	Tüm dosyayı okur, sonucu liste olarak döndürür.

Grafik Çizimleri

Turtle (Kaplumbağa) modülü ile çizimler:

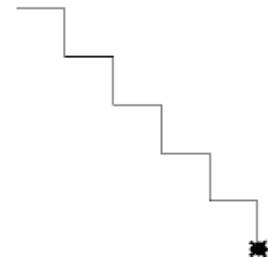
Örnek 5: Kare çizen Python programı.

```
from turtle import *
kenar=100
forward(kenar) #kenar kadar ilerle
left(90) #90 derece s.i.t. yönünde dön
forward(kenar)
left(90)
forward(kenar)
left(90)
forward(kenar)
```



Örnek 6: Merdiven çizimi.

```
from turtle import *
shape('turtle')
for i in range(5):
    forward(50)
    right(90)
    forward(50)
    left(90)
```

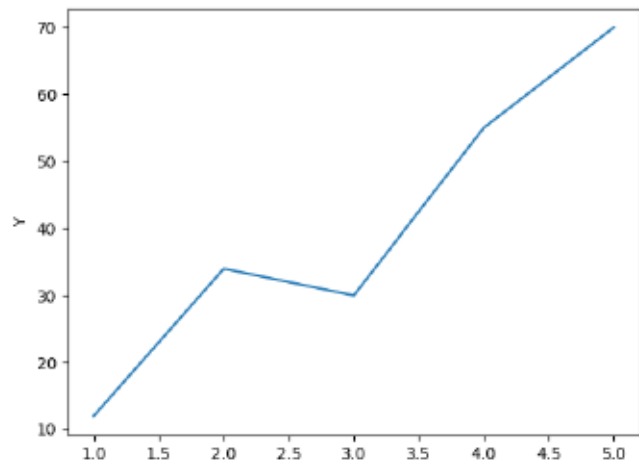


Matplotlib Modülü ile Grafik Çizimi:

Örnek 1: Verilen nokta çiftlerinin xy grafiğini çizen Python programını yazınız.

(1,12), (2,34), (3,30), (4,55), (5,70)

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
y=[12,34,30,55,70]
plt.plot(x,y)
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



NumPy Modülü: Kullanılmadan önce bilgisayarınıza yüklenmelidir

0 ile 10 arasındaki sayıları içeren listenin oluşturulması:

```
import numpy as np  
x=np.arange(0,11,1)  
print(x)
```

} [0 1 2 3 4 5 6 7 8 9 10]

x listesinin fonksiyonda yerine konularak fx listesinin oluşturulması:

```
fx=3*x**2-2*x+4  
print(fx)
```

} [4 5 12 25 44 69 100 137 180 229 284]

x=np.**arange**(0,11,1) : 0 ile 10 arasında 1'er artarak liste şeklinde sayı üretilmesi
VEYA

x=np.**linspace**(0,10,15) : 0 ile 10 arasında 15 adet eşit aralıklı sayı üretilmesi

0 2 4 6 8 10