

LİSTELER

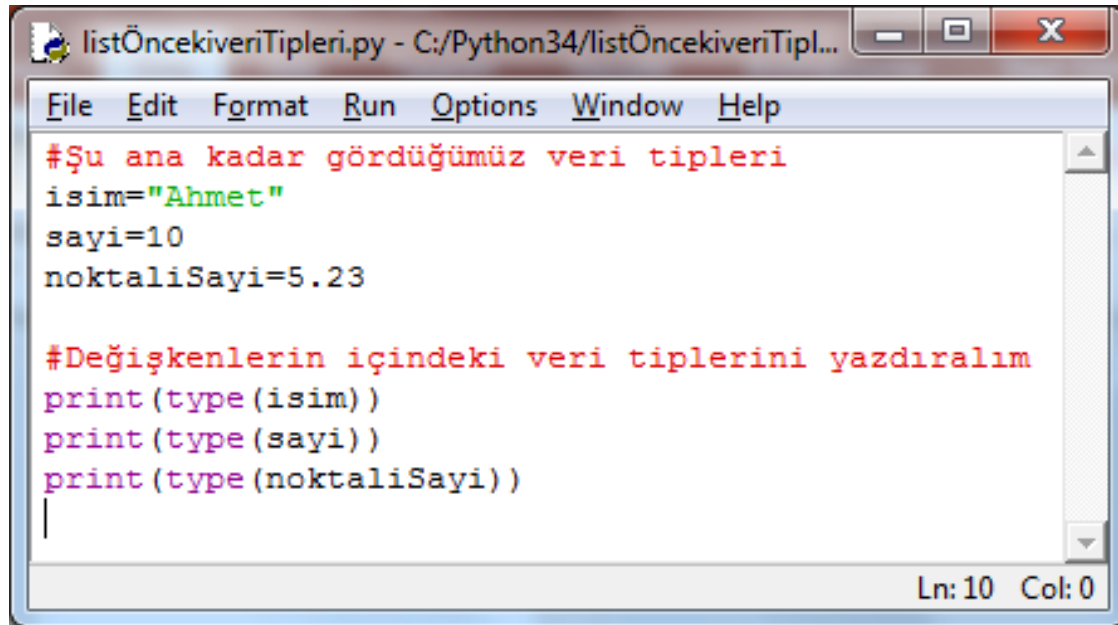
Liste veri tipi nedir?

Liste tanımlama

Liste Metodları

Şu ana kadar gördüğümüz veri tipleri

Şu ana kadar «int», «float» sayı tipleri ve «string» karakter dizisi türünde değişken tiplerini gördük.



```
listÖncekiveriTipleri.py - C:/Python34/listÖncekiveriTipl...
File Edit Format Run Options Window Help
#Şu ana kadar gördüğümüz veri tipleri
isim="Ahmet"
sayi=10
noktaliSayi=5.23

#Değişkenlerin içindeki veri tiplerini yazdıralım
print(type(isim))
print(type(sayi))
print(type(noktaliSayi))
|
Ln: 10 Col: 0
```

```
<class 'str'>
<class 'int'>
<class 'float'>
>>>
```

LİSTE VERİ TİPİ

Şu ana kadar değişkenlere sadece bir değer tutabiliyorduk. Listeler sayesinde bir değişken içerisinde birden fazla ve farklı tiplerde değer tutabiliriz.

Örnek:

```
listemiz=["ahmet","ayşe",10,30,-20,3.25]
```

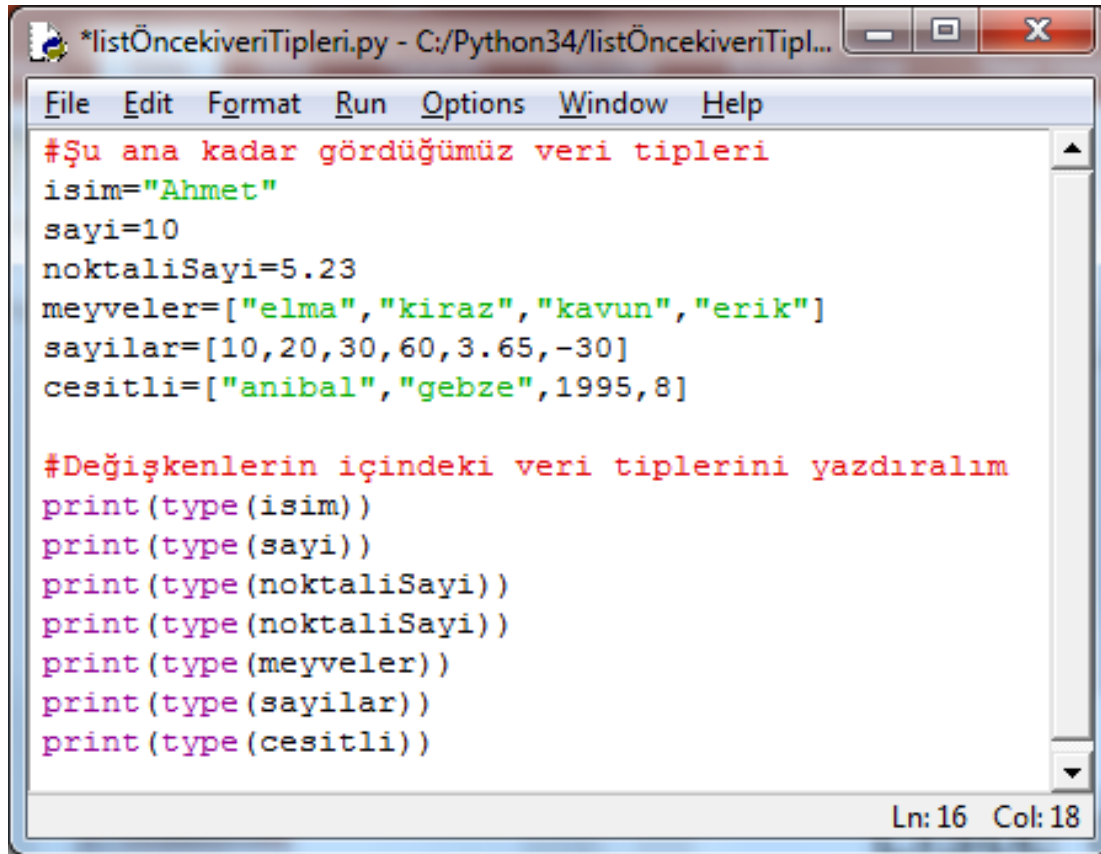
Liste oluşturulurken öğeler [] içine ve virgülle ayrılarak yazılır
Listemiz isimli değişkende 6 adet veri yer almaktadır.

Liste özelliğindeki bir değişkenin tipini öğrenelim

```
listemiz=["ahmet","ayşe",10,30,-20,3.25]  
print(type(listemiz))
```

```
<class 'list'>  
>>> |
```

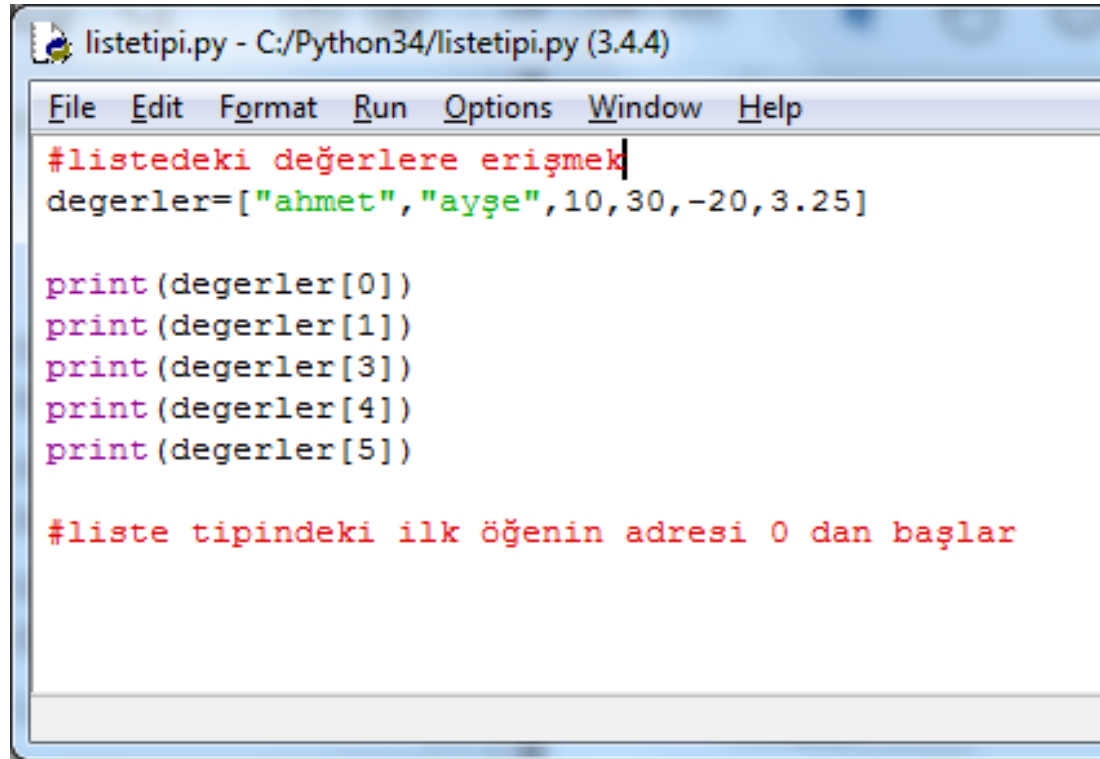
Değişkenlere atanan veri tiplerini öğrenelim



```
*listÖncekiveriTipleri.py - C:/Python34/listÖncekiveriTipl...  
File Edit Format Run Options Window Help  
#Şu ana kadar gördüğümüz veri tipleri  
isim="Ahmet"  
sayi=10  
noktaliSayi=5.23  
meyveler=["elma", "kiraz", "kavun", "erik"]  
sayilar=[10, 20, 30, 60, 3.65, -30]  
cesitli=["anibal", "gebze", 1995, 8]  
  
#Değişkenlerin içindeki veri tiplerini yazdıralım  
print(type(isim))  
print(type(sayi))  
print(type(noktaliSayi))  
print(type(noktaliSayi))  
print(type(meyveler))  
print(type(sayilar))  
print(type(cesitli))  
Ln: 16 Col: 18
```

```
<class 'str'>  
<class 'int'>  
<class 'float'>  
<class 'float'>  
<class 'list'>  
<class 'list'>  
<class 'list'>
```

Listedeki verilere erişmek



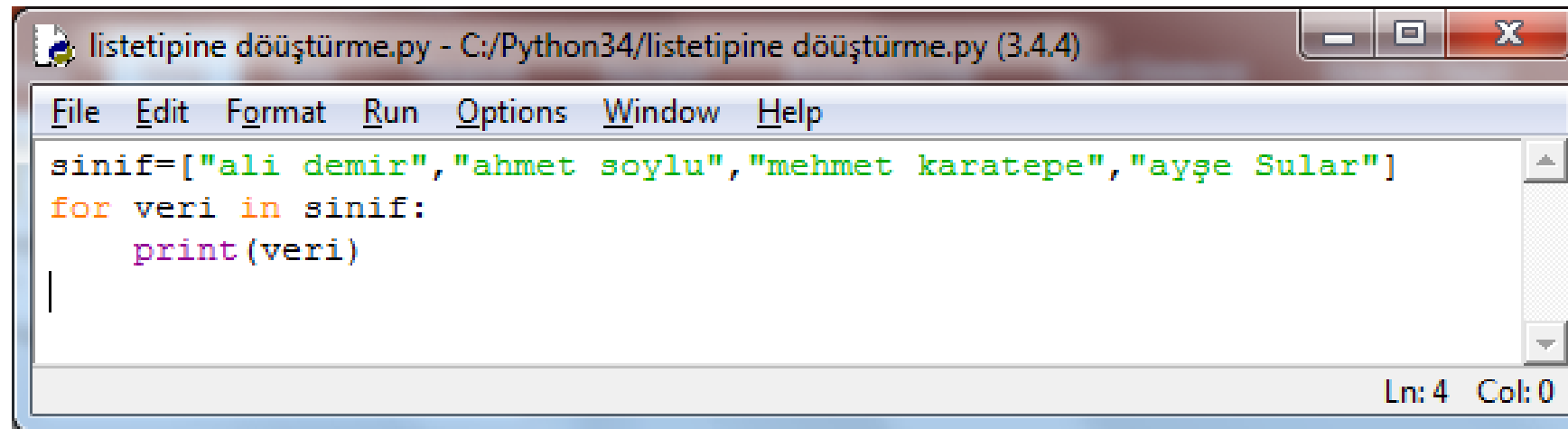
```
listetipi.py - C:/Python34/listetipi.py (3.4.4)
File Edit Format Run Options Window Help
#listedeki değerlere erişmek
degerler=["ahmet", "ayşe", 10, 30, -20, 3.25]

print(degerler[0])
print(degerler[1])
print(degerler[3])
print(degerler[4])
print(degerler[5])

#liste tipindeki ilk öğenin adresi 0 dan başlar
```

```
ahmet
ayşe
30
-20
3.25
>>> |
```

For döngüsü ile liste elemanlarını almak



The screenshot shows a Python IDE window titled "listetipine dönüştürme.py - C:/Python34/listetipine dönüştürme.py (3.4.4)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python code:

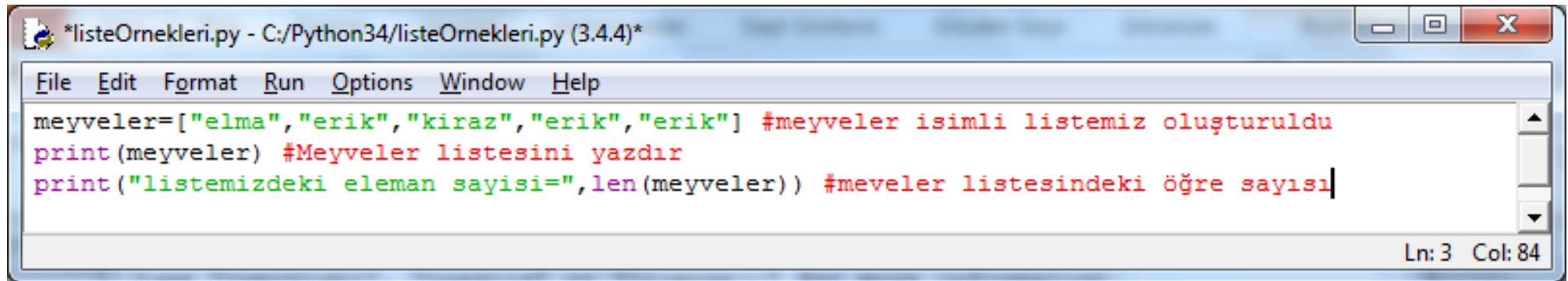
```
sinif=["ali demir","ahmet soylu","mehmet karatepe","ayşe Sular"]
for veri in sınıf:
    print(veri)
|
```

The status bar at the bottom right indicates "Ln: 4 Col: 0".

```
ali demir
ahmet soylu
mehmet karatepe
ayşe Sular
>>> |
```

Listedeki öge sayısını öğrenmek

Daha önceden gördüğümüz `len()` fonksiyonunu listemizdeki eleman sayısını öğrenmek için kullanabiliriz



```
*listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)*
File Edit Format Run Options Window Help
meyveler=["elma","erik","kiraz","erik","erik"] #meyveler isimli listemiz oluřturuldu
print(meyveler) #Meyveler listesini yazdır
print("listemizdeki eleman sayısı=",len(meyveler)) #meyveler listesindeki öğre sayısı
Ln: 3 Col: 84
```

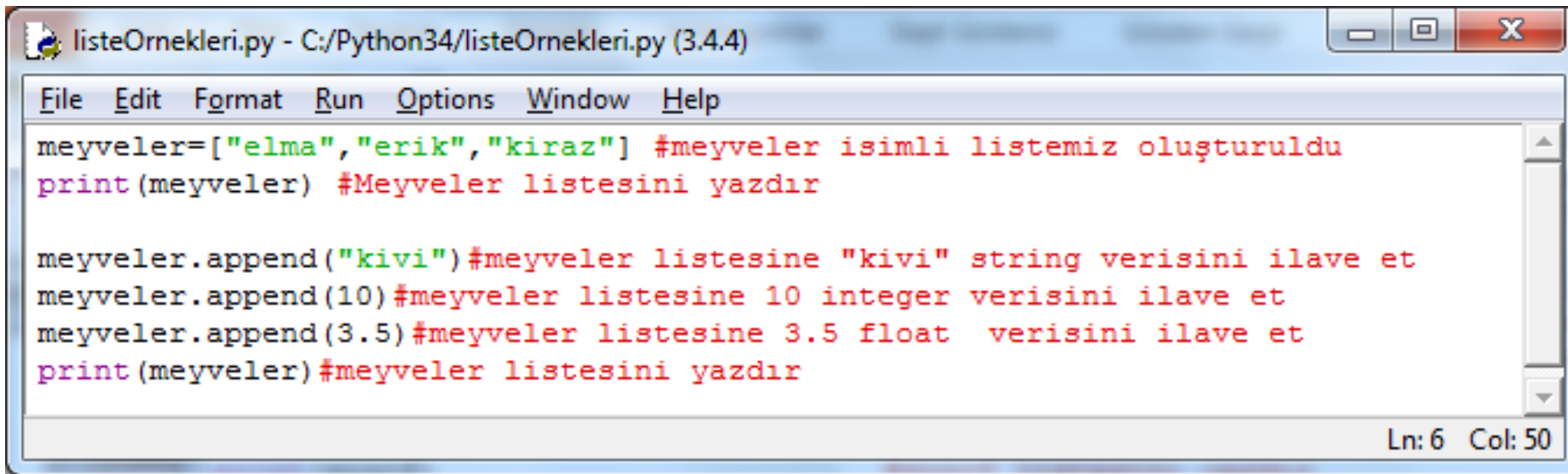
```
----- RESTART: C:/Python34/113
['elma', 'erik', 'kiraz', 'erik', 'erik']
listemizdeki eleman sayısı= 5
>>> |
```

Listelerin Metotları

Liste metodlarını kullanım biçimi
`listeismi.metodismi(parametreler)`

append(veri) metodu

- append kelimesi İngilizcede 'eklemek, ilave etmek, iliştmek' gibi anlamlara gelir. Ögeyi listenin sonuna ekler.

A screenshot of a Python IDE window titled 'listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)'. The window contains a Python script that demonstrates the use of the 'append' method on a list. The script defines a list 'meyveler' with fruit names, prints it, then appends 'kivi', 10, and 3.5 to it, and prints the updated list. The status bar at the bottom right shows 'Ln: 6 Col: 50'.

```
File Edit Format Run Options Window Help
meyveler=["elma","erik","kiraz"] #meyveler isimli listemiz oluřturuldu
print(meyveler) #Meyveler listesini yazdır

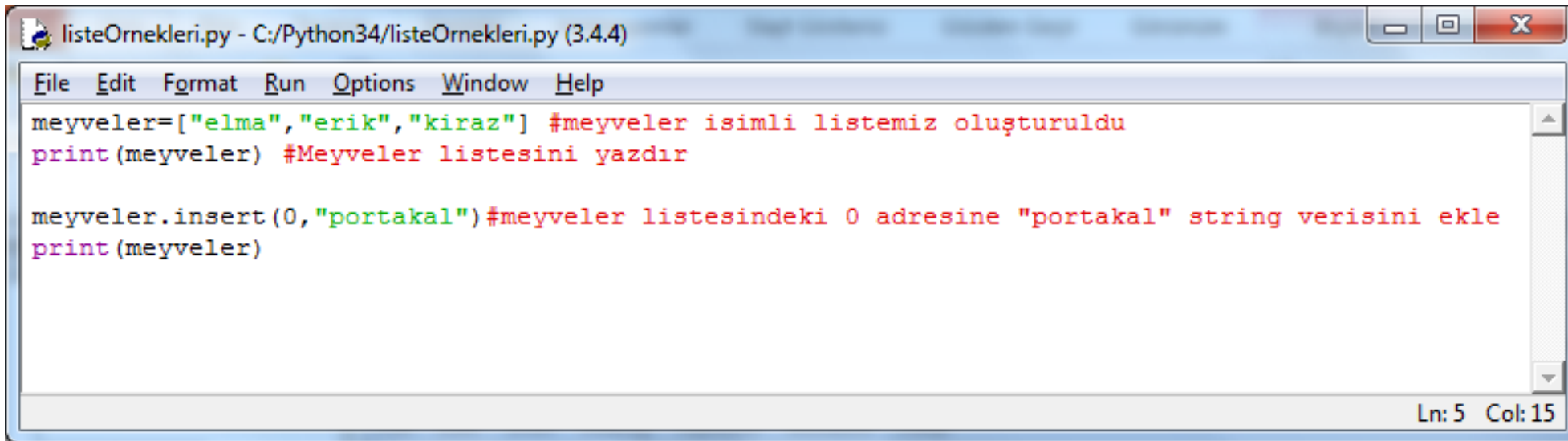
meyveler.append("kivi")#meyveler listesine "kivi" string verisini ilave et
meyveler.append(10)#meyveler listesine 10 integer verisini ilave et
meyveler.append(3.5)#meyveler listesine 3.5 float verisini ilave et
print(meyveler)#meyveler listesini yazdır

Ln: 6 Col: 50
```

```
['elma', 'erik', 'kiraz']
['elma', 'erik', 'kiraz', 'kivi', 10, 3.5]
```

insert(konum,veri) metodu

Listedeki istenilen konuma öge eklemenizi sağlar



```
listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)
File Edit Format Run Options Window Help
meyveler=["elma","erik","kiraz"] #meyveler isimli listemiz olusturuldu
print(meyveler) #Meyveler listesini yazdır

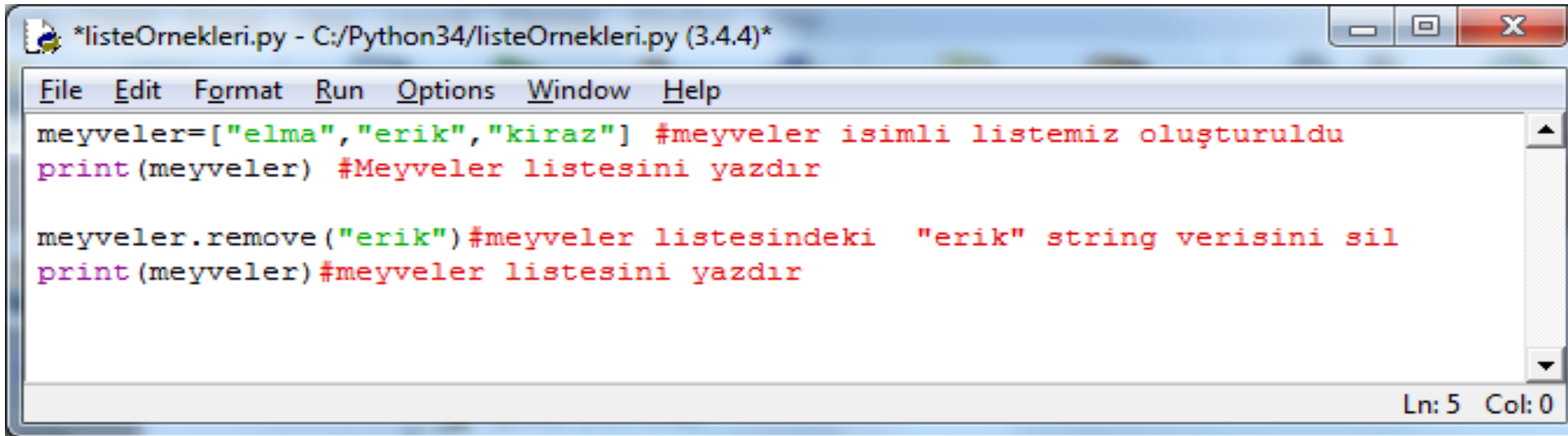
meyveler.insert(0,"portakal") #meyveler listesindeki 0 adresine "portakal" string verisini ekle
print(meyveler)
```

Ln: 5 Col: 15

```
['elma', 'erik', 'kiraz']
['portakal', 'elma', 'erik', 'kiraz']
>>> |
```

remove(veri) metodu

Listedeki belirtilen ilk veriyi siler. İlgili eleman bulunmaz ise hata verir.

A screenshot of a Python IDE window titled '*listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)*'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The main text area contains the following Python code:

```
meyveler=["elma","erik","kiraz"] #meyveler isimli listemiz oluřturuldu
print(meyveler) #Meyveler listesini yazdır

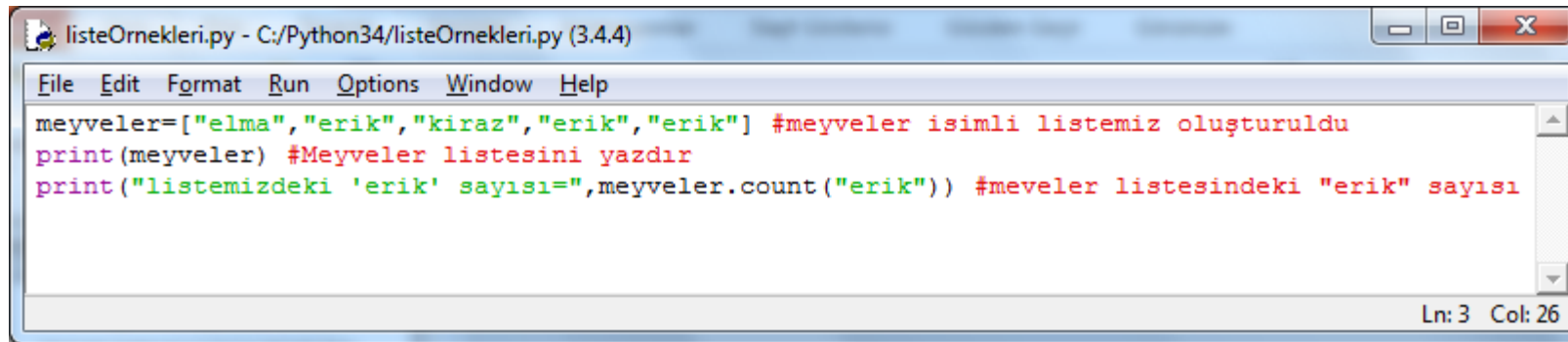
meyveler.remove("erik") #meyveler listesindeki "erik" string verisini sil
print(meyveler) #meyveler listesini yazdır
```

The status bar at the bottom right shows 'Ln: 5 Col: 0'.

```
['elma', 'erik', 'kiraz']
['elma', 'kiraz']
>>> |
```

count(veri) metodu

- count() metodu da bir ögenin o veri tipi içinde kaç kez geçtiğini belirtir

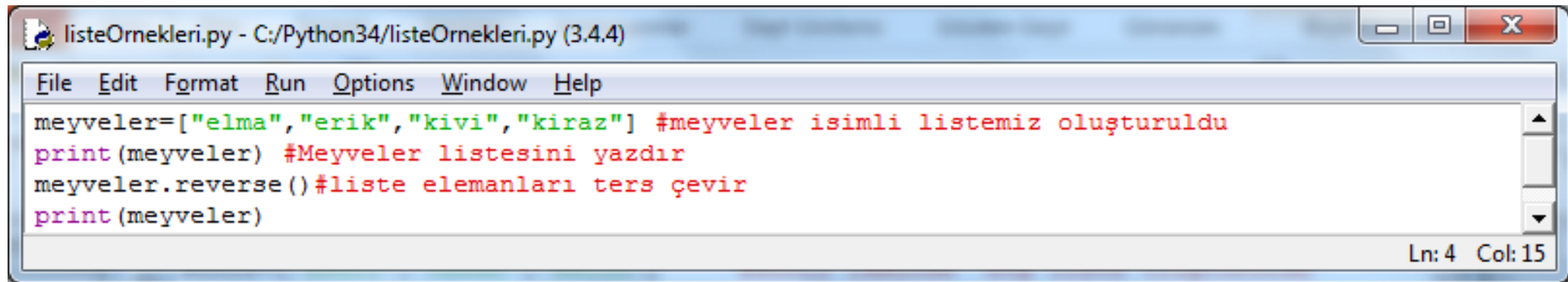


```
listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)
File Edit Format Run Options Window Help
meyveler=["elma","erik","kiraz","erik","erik"] #meyveler isimli listemiz oluřturuldu
print(meyveler) #Meyveler listesini yazdır
print("listemizdeki 'erik' sayısı=",meyveler.count("erik")) #meyveler listesindeki "erik" sayısı
Ln: 3 Col: 26
```

```
['elma', 'erik', 'kiraz', 'erik', 'erik']
listemizdeki 'erik' sayısı= 3
>>> |
```

reverse() metodu

- Listedeki elemanların dizilişini ters çevirir.



```
listeOrnekleri.py - C:/Python34/listeOrnekleri.py (3.4.4)
File Edit Format Run Options Window Help
meyveler=["elma","erik","kivi","kiraz"] #meyveler isimli listemiz oluřturuldu
print(meyveler) #Meyveler listesini yazdır
meyveler.reverse() #liste elemanları ters çevir
print(meyveler)
Ln: 4 Col: 15
```

```
['elma', 'erik', 'kivi', 'kiraz']
['kiraz', 'kivi', 'erik', 'elma']
>>> |
```

sort() metodu

- Listeyi artan biçimde sıralar(a-z)

```
File Edit Format Run Options Window Help
isimler=["zeynep","fatma","ayşe","kadir","hasan","buse","ceren"] #isimler adında liste oluşturuldu
sayilar=[2,10,5,4,3,60,100,40,12] #sayilar adında liste oluşturuldu
print(isimler) #İsimler listesini yazdır
print(sayilar) #sayilar listesini yazdır
isimler.sort()#isimler adlı listeyi sırala
sayilar.sort()#sayilar adlı listeyi sırala
print(isimler)
print(sayilar)|
```

```
['zeynep', 'fatma', 'ayşe', 'kadir', 'hasan', 'buse', 'ceren']
[2, 10, 5, 4, 3, 60, 100, 40, 12]
['ayşe', 'buse', 'ceren', 'fatma', 'hasan', 'kadir', 'zeynep']
[2, 3, 4, 5, 10, 12, 40, 60, 100]
```

Listede kullanılan metodlar

Metod	açıklama
append(veri)	Listenin sonuna girilen veriyi ilave eder
insert(konum,veri)	Listede istenilen konuma veri eklemenizi sağlar
remove(veri)	Liste içindeki belirtilen ilk veriyi siler
count(veri)	Belirtilen verinin liste içindeki sayısını verir
reverse()	Liste öğelerini tersine çevirir
sort()	Liste öğelerini büyükten küçüğe yada a-z ye sıralar

Örnek: Boş liste tanımlayıp içerisine isim ekleme ve yazdırma

```
sinifListemiz=[]
while True:
    isim=input("İsim giriniz çıkmak için q karfine basınız:")
    if isim=="q":
        break
    sınıfListemiz.append(isim)

print(sınıfListemiz)
```

```
classmate.py
İsim giriniz çıkmak için q karfine basınız:hasan
İsim giriniz çıkmak için q karfine basınız:kadir
İsim giriniz çıkmak için q karfine basınız:ali
İsim giriniz çıkmak için q karfine basınız:ayşe
İsim giriniz çıkmak için q karfine basınız:fatma
İsim giriniz çıkmak için q karfine basınız:selim
İsim giriniz çıkmak için q karfine basınız:sude
İsim giriniz çıkmak için q karfine basınız:q
['hasan', 'kadir', 'ali', 'ayşe', 'fatma', 'selim', 'sude']
```

Listeler için kullanılabilen random modülü fonksiyonları

```
from random import *
sinif=["ali","ayşe","fatma","kadir","salih","veli"]
print(choice(sinif))#listemizden rastgele bir öğe seçer
print(sample(sinif,3))#listemizden rastgele 3 öğe seçer
shuffle(sinif)#listemizdeki öğeleri karıştırır
print(sinif)#listemizin öğelerinin karışmış hali
```

```
fatma
['kadir', 'veli', 'ayşe']
['kadir', 'salih', 'ali', 'veli', 'ayşe', 'fatma']
```

Turtle, random modülünü ve listeleri kullanıldığı bir örnek

```
direrandom.py - C:/Python34/direrandom.py (3.4.4)
File Edit Format Run Options Window Help
from turtle import * #turtle modülünü imoport et
from random import * #random modülünü import et
#renk listesi tanımla
renkler=["blue","yellow","red","purple","gray","brown","pink","green"]
speed(50) #hızı 50 yap
for i in range(20):
    cizgirengi=choice(renkler) #listeden bir renk sec
    dolgurengi=choice(renkler) #listeden bir renk seç
    color(cizgirengi,dolgurengi) #renk ayarla
    penup() #kalemi kaldır
    x=randint(-300,300) #rastgele -300 ile 300 arasında tamsayı üret
    y=randint(-300,300) #rastgele -300 ile 300 arasında tamsayı üret
    goto(x,y) #belirtilen x ve y konumuna git
    birim=randint(50,100) #50 ile 100 arasında tamsayı üret
    pendown() #kalemi bastır
    begin_fill() #boyamaya başla
    circle(birim) #birim ölçüsünde daire çiz
    end_fill() #boyamayı bitir
Ln: 2 Col: 47
```

