Listeye Eleman Ekleme

Python listelerinin sonuna bir eleman eklemek için append() metodu kullanılır.

Örnek

liste = ["banana", "grape", "cherry"]

liste.append("apple")

print(liste) # ["banana", "grape", "cherry","apple"]

Python listelerinde belirtilen bir indeks'e eleman eklemek için insert() metodu kullanılır.

liste = ["banana", "grape", "cherry"]

liste.insert(2,"apple")

print(liste) # ["banana", "grape", "apple","cherry"]

Listeden Eleman Silme

Python listelerinden eleman silmek için kullanabileceğimiz farklı metotlar mevcuttur.

Listeden bir eleman silmek için remove() metodunu kullanabiliriz.

liste = ["banana", "grape", "cherry"]

liste.remove("grape")

print(liste) # ["banana","cherry"]

Python listelerinde belirtilen bir indeks' deki elemanı silmek için pop() metodu kullanılır. Eğer indeks numarası belirtmezsek listenin son elemanı silinir.

liste = ["banana", "grape", "cherry"]

liste.pop(1)

print(liste) # ["banana","cherry"]

İndeks numarası vermediğimizde ise son eleman silinir.

liste = ["banana", "grape", "cherry"]

liste.pop()

print(liste) # ["banana", "grape"]

del() metodu ile her hangi bir indeks numarasındaki elemanı silebiliriz.

liste = ["banana", "grape", "cherry"]

del liste[2]

print(liste) # ["banana", "grape"]

Eğer ki del() metoduna indeks numarası vermezsek listeyi olduğu gibi siler.

liste = ["banana", "grape", "cherry"]

del liste

print(liste) # NameError: name 'list' is not defined

Bu durumda liste objesine ulaşmak istediğimizde NameError alırız ve açıklama da ise list tanımlanmadı şeklinde hata mesajı gelir.

Ayrıca del komutu ile listeyi sildiğimiz gibi clear() metodu ile de listeyi silebiliriz. Ancak arada ki fark del ile obje referansıda silindiğinde listeye ulaşmak istediğimizde NameError alırken clear() metodu ile hata almayız çünkü listenin referansı bellekte olmaya devam eder ancak içi boş olur.

liste = ["banana", "grape", "cherry"]

liste.clear()

print(liste) # []

Gördüğünüz gibi print(liste) ile bize gelen değer [ ] boş liste tanımlamasıdır. Bellekte halen yer tutar ve bu liste üzerine tekrar eleman eklemeye devam edebiliriz.

Liste Kopyalama

List bir class ve bellekte referans tip olarak ele alınır dolayısıyla bir listeyi başka bir listeye atamak istediğimizde liste elamanları kopyalanmaz bunun yerine listenin bellekteki adres bilgisi kopyalanır.

a = ["apple","banana"]

b = ["grape","cherry"]

a = b

Burada b'nin adresi a listesine atanmıştır. Dolayısıyla artık a ve b listeleri belleğin aynı adresindeki aynı verilere sahiptir. (["grape","cherry"])

Dolayısıyla a ya da b üzerinde yaptığımız her hangi bir değişiklik iki liste üzerinde de yapılmış olur.

a = ["apple","banana"]

b = ["grape","cherry"]

a = b

b[0] = "updated"

print(a, b) # çıktı: ['updated', 'cherry'] ['updated', 'cherry']

Dolayısıyla listeleri kopyalarken kullanmamız gereken bazı liste metotları vardır.

Bir liste içeriğini başka bir listeye atamak için copy() metodunu kullanabiliriz.

a = ["apple","banana"]

b = ["grape","cherry"]

a = b.copy()

b[0] = "updated"

print(a, b) # çıktı: ['grape', 'cherry'] ['updated', 'cherry']

Gördüğünüz gibi atama işleminden sonra b[0] indeks üzerinde yaptığımız bir güncelleme artık a listesini etkilemedi çünkü iki listede farklı adreslere sahip birer objedir burada adres kopyalaması değil de sadece içlerindeki bilgiler kopyalanmıştır.

Liste kopyalamak için kullanabileceğimiz bir başka metor ise list() metodudur.

a = ["apple","banana"]

b = ["grape","cherry"]

a = list(b)

b[0] = "updated"

print(a, b) # çıktı: ['grape', 'cherry'] ['updated', 'cherry']

Liste Elemanlarını Sıralama

Liste elemanlarını sıralamak için sort() metodu kullanılır.

numbers = [1, 10, 5, 16, 4, 9, 10]

letters = ['a', 'g', 's', 'b', 'y', 'a', 's']

numbers.sort() # sayısal büyüklük olarak küçükten büyüğe sıralanır.

letters.sort() # alfabetik olarak a-z' e doğru sıralanır.

Tersten sıralamak istediğimizde ise reverse=True parametresini gönderebiliriz.

numbers.sort(reverse=True) # sayısal büyüklük olarak büyükten küçüğe sıralanır.

letters.sort(reverse=True) # alfabetik olarak z-a'ya doğru sıralanır.

Liste elemanlarını reverse() metodu ile tersten yazdırabiliriz.

numbers = [1, 10, 5]

letters = ['a', 'g', 's']

numbers.reverse() # [5, 10, 1]

letters.reverse() # ['s', 'g', 'a']

Diğer Liste Metotları

Min() ve Max() metodu

Bir liste içerisindeki minimum ve maksimum değeri ister sayısal ister alfabetik olarak alabiliriz.

numbers = [1, 10, 5, 16, 4, 9, 10]

letters = ['a', 'g', 's', 'b', 'y', 'a', 's']

min(numbers) # 1

max(numbers) # 16

max(letters) # y

min(letters) # a

Count() metodu

Bir liste içerisindeki tekrarlayan elemanların sayısını almak için count() metodunu kullanırız.

numbers = [1, 10, 5, 16, 4, 9, 10]

letters = ['a', 'g', 's', 'b', 'y', 'a', 's']

numbers.count(10) # 2

letters.count('a') # 2

Python Liste Metot Uygulamaları

names = ['Ali','Yağmur','Hakan','Deniz']

years = [1998, 2000, 1998, 1987]

1- "Cenk" ismini listenin sonuna ekleyiniz.

names.append('Cenk')

2- "Sena" değerini listenin başına ekleyiniz.

names.insert(0, 'Sena')

names.insert(-1, 'Mehmet')

names.insert(len(names), 'Mehmet')

3- "Deniz" ismini listeden siliniz.

names.remove('Deniz')

names.pop()

names.pop(1)

4- "Deniz" isminin indeksi nedir ?

index = names.index('Deniz')

names.pop(index)

5- "Ali" listenin bir elemanı mıdır ?

result = 'Ali' in names

result = names.index('Ali')

6- Liste elemanlarını ters çevirin.

names.reverse()

7- Liste elemanlarını alfabetik olarak sıralayınız.

names.sort()

8- years listesini rakamsal büyüklüğe göre sıralayınız.

years.sort()

9- str = "Chevrolet,Dacia" karakter dizisini listeye çeviriniz.

str = "Chevrolet,Dacia"

result = str.split(',')

10- years dizisinin en büyük ve en küçük elemanı nedir ?

min = min(years)

max = max(years)

print(min, max)

11- years dizisinde kaç tane 1998 değeri vardır ?

result = years.count(1998)

12- years dizisinin tüm elemanlarını siliniz.

years.clear()

13- Kullanıcıdan alacağınız 3 tane marka bilgisini bir listede saklayınız.

markalar = []

marka = input("marka: ")

markalar.append(marka)

marka = input("marka: ")

markalar.append(marka)

marka = input("marka: ")

markalar.append(marka)

print(markalar)