LİSTELER

Liste veri toplama aktarma yollarından biridir. Python'da dört çeşit veri toplama şekli vardır bunlardan biri liste yöntemidir. Liste veri türü sıralı ve değiştirilebilir olmaktadır eleman çıkarılabilir yenisi gelebilir veya üzerinde oynama yapılabilir. Tuple veri türünde sıralı ve değiştirilemez veya değişmez bir türdür. Set veri türünde sıralanmamış, indekslenmemiş ve değiştirilemeyen bir koleksiyondur. Yeni üye eklenebilir fakat yinelemelere izin vermez. Sözlük veri türü Sırasız, değiştirilebilir(değiştirilebilir) ve indekslenmiş bir türdür. Arkadaşlar aslında veri türlerinde gördüğümüz terimler bizlere yabancı gelmedi bir de listeleme detaylarını inceleyip detaylarını öğrenmeye başlayalım. Nasıl liste oluşturul ile başlayalım.

## Python'da Liste Nasıl Oluşturulur

İki şekilde liste oluşturabiliriz ikisi de aynı görevi üstelnmektedir hangisi kolayınıza gelirse onu tercih edebilirsiniz ben ikisinide sizlere göstermek istiyorum.

* Liste yerleşik işlev()

lst  =  list()

* Köşeli parantez []

lst  =[]

Aşağıdaki liste örneklerini inceleyelim. Farklı veri türleri bir listede bulunabilir örneği incelerken bu bilgi aklımızda bulunsun.

şehirler  =['İstanbul', 'Adana  ' , 'Samsun','İzmir','Ordu']

market\_listesi=['süt','yağ','un','yumurta','peynir','sucuk']

gezilecek\_yerler=['Sultanahmet','Ayasofya','Kız Kulesi','Galata Kulesi']

print('şehirler:',şehirler)

print('alınacaklar:',market\_listesi)

print('gezilecek yerler:',gezilecek\_yerler)

Çıktı:

şehirler: ['İstanbul', 'Adana  ', 'Samsun', 'İzmir', 'Ordu']

alınacaklar: ['süt', 'yağ', 'un', 'yumurta', 'peynir', 'sucuk']

gezilecek yerler: ['Sultanahmet', 'Ayasofya', 'Kız Kulesi', 'Galata Kulesi']

## Python'da Pozitif İndeksleme Kullanarak Liste Öğelerine Erişme

Daha önce de bahsettiğimiz gibi programlama da 0’dan başlarız ve liste indeksleme için 0'dan başlayarak pozitif indeksleme gerçekleştirim.

Örnek:

dil  =  ['Python','c++','R','SQL']

ılk\_program= dil [ 0 ]

print (ılk\_program) # Python

ikinci\_program =  dil [ 1 ]

print ( ikinci\_program) # c++

üçüncü\_program=dil[2]

print(üçüncü\_program) # R

son\_hal =  len ( dil ) -  1

son\_hal =  dil [ son\_hal ]

print ( son\_hal ) # SQL

## Pythpn'da Negatif İndeksleme Kullanarak Liste Öğelerine Erişme

Negatif indeksleme, sondan başlamak anlamına gelir, -1 son öğeyi, -2 sondan ikinci öğeyi ifade eder. Burada listedeki son eleman -1, listedeki ilk eleman -x şeklinde yazılır. Aşağıda örnekte göreceğiniz gibi son eleman -1 ilk elman -4 negatif indeks kullanılarak listeleme yapılmıştır.

dil  =  ['Python','c++','R','SQL']

ılk\_program= dil [ -4 ]

print (ılk\_program) # Python

ikinci\_program =  dil [ -3 ]

print ( ikinci\_program) ) # c++

üçüncü\_program=dil[-1]

print(üçüncü\_program) ) # SQL

## Python'da Liste Öğelerini Açma

Liste açma örneklerimizi aşağıdaki şekilde inceleyelim.

lst  = [ 'madde1' , 'madde2' , 'madde3'  ]

ılk\_program, ikinci\_program,üçüncü\_program   =  lst

print( ılk\_program)    # madde1

print( ikinci\_program )   #madde2

print ( üçüncü\_program)   # madde3

ÖRNEK:

ülke = ['Türkiye', 'Fransa','Belçika','Suriye','Danimarka','Finlandiya','Norveç','Almanya','Estonya']

tr, fr, bç, su, \*scandic, es = ülke

print(tr)

print(fr)

print(bç)

print(su)

print(scandic)

print(es)

ÇIKTI:

Türkiye

Fransa

Belçika

Suriye

['Danimarka', 'Finlandiya', 'Norveç', 'Almanya']

Estonya

## Python'da Bir Listeden Öğeleri Dilimleme

* Pozitif İndeksleme: Başlangıç, bitiş ve adımı belirterek bir dizi pozitif indeks belirleyebiliriz, dönüş değeri yeni bir liste olacaktır.

fruits = ['banana', 'orange', 'mango', 'lemon']

all\_fruits = fruits[0:4]

print(all\_fruits) # ['banana', 'orange', 'mango', 'lemon']

ll\_fruits = fruits[0:]

print(ll\_fruits) # ['banana', 'orange', 'mango', 'lemon']

orange\_and\_mango = fruits[1:3]

print(orange\_and\_mango)#['orange', 'mango']

orange\_mango\_lemon = fruits[1:]

print(orange\_mango\_lemon)# ['orange', 'mango', 'lemon']

orange\_and\_lemon = fruits[::2] # Burada orange ve lemon hariç diğer iki değişkeni alsın diyor

print(orange\_and\_lemon)#['banana', 'mango']

* Negatif İndeksleme: Başlangıç, bitiş ve adımı belirterek bir dizi negatif indeks belirleyebiliriz, dönüş değeri yeni bir liste olacaktır.

fruits = ['banana', 'orange', 'mango', 'lemon']

all\_fruits = fruits[-4:]

print(all\_fruits) # ['banana', 'orange', 'mango', 'lemon']

orange\_and\_mango = fruits[-3:-1]

print(orange\_and\_mango)#['orange', 'mango']

orange\_mango\_lemon = fruits[-3:]#-3'den

print(orange\_mango\_lemon)#['orange', 'mango', 'lemon']

## Python'da Listeleri Değiştirme

Liste, değiştirilebilir veya değiştirilebilir sıralı bir öğe koleksiyonudur. Meyve listesini değiştirelim içerisine farklı meyve isimleri yazalım. Örneğimiz deki değiştirme yöntemini inceleyelim.’0’ “muz” değişkenini “avokado” yerine değiştirdiğimizi ilk satırda görebiliriz. Daha sonra ’1’ “portakal” değişkenini “elma” yerine değiştirdiğimizi ikinci satırda görebiliriz.

meyveler  = [ 'muz' , 'portakal' , 'mango' , 'limon' ]

meyveler [ 0 ] =  'avokado'

print ( meyveler )        # ['avokado', 'portakal', 'mango', 'limon']

meyveler [ 1 ] =  'elma'

print( meyveler )        # ['avokado', 'elma', 'mango', 'limon']

son\_değer =len( meyveler ) -  1

meyveler [ son\_değer] = 'kireç'

print( meyveler )         # ['avokado', 'elma', 'mango', 'kireç']

## Python'da Listedeki Öğeleri Kontrol Etme

Burada yoklama alıyormuş gibi düşünebiliriz. Burada liste içerisinde o değişkenin bulunup bulunmadığını sorgulayacağız ve "True" ya da "False" sonucunu alacağız. Örneğimizi inceleyelim.

meyve = ['muz', 'portakal','mango']

var\_mı = 'muz' in meyve

print(var\_mı)  # True

var\_mı = 'mango' in meyve

print(var\_mı)  # True

## Python'da Listeye Öğe Ekleme

* Var olan bir listenin sonuna değişken eklemek için append() komutu kullanırız öğrendiğimizi inceleyelim.

meyve = ['muz', 'portakal','mango']

meyve.append('karpuz')

print(meyve)#['muz', 'portakal', 'mango', 'karpuz']

* Var olan bir listeye istediğimiz yere yeni değişken eklemek için insert() komutu kullanılır. Nasıl kullanıldığını örneğimizde inceleyelim. meyve.insert(2, ‘karpuz’) komutumuzda 1-muz , 2-portakal dolayısıyla portakaldan sonra karpuz yerleştiririm çünkü komutta 2 yazdım umarım anlaşılır oluyordur aşağıdaki örneği inceleyelim.

meyve = ['muz', 'portakal','mango']

meyve.insert(2, 'karpuz')

print(meyve)#['muz', 'portakal', 'karpuz', 'mango']

meyve.insert(3,'kavun')

print(meyve)#['muz', 'portakal', 'karpuz', 'kavun', 'mango']

## Python'da Listeden Öğeleri Kaldırma

Listeye ekleme işlemlerimizi gördük şimdi listeden kaldırma ile ilgili komutumuz olan remove() komutunu ele alalım ve basit bir örneği inceleyelim.

meyve = ['muz', 'portakal','mango']

meyve.remove('muz')

print(meyve)#['portakal', 'mango']

## Python'da pop Kullanarak Öğeleri Kaldırma

pop() komutu parantez içerisine yazacağımız pozitif eleman sayısı 0 1 2 3 hangisini çıkarmak istersek onu yazarız. Örneğin ‘portakal’ değişkeni çıkarmak istiyorum pop(1) yazarak ‘portakal’ değişkeni listeden çıkar.

meyve=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

meyve.pop(1)

print(meyve)#['muz', 'karpuz', 'kavun', 'mango']

## Python'da Del Kullanarak Öğeleri Kaldırma

Del belirtilen değişkeni ortadan kaldırır. Hadi örneğimize bakalım ve pekiştirelim.

meyve=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

del meyve[0]

print(meyve)#['portakal', 'karpuz', 'kavun', 'mango']

del meyve[2]

print(meyve)['portakal', 'karpuz', 'mango'] muz silindikten sonra

portakal 0 karpuz 1 kavun 2 olur ve kavun silinir.

## Python'da Liste Öğelerini Temizleme

clear() komutu ile liste temizleme işlemini gerçekleştiririm.

meyve=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

meyve.clear()

print(meyve)#[]

## Python'da Liste Kopyalama

Herhangi bir istediğim listeyi başka bir değişkene tanımlamak için copy() komutu kullanabilirim aşağıda örneğimizi inceleyelim.

Sebze değişkeni tanımladım daha sonra sebzenin içine meyve yazılmış ben de sebze dizisini içini silip daha sonra listeyi meyve değişkenine aktardım.

sebze=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

meyve=sebze.copy()

print(meyve)#['muz', 'portakal', 'karpuz', 'kavun', 'mango']

sebze.clear()

print(sebze)#[]

## Python'da Liste Birleştirme

İki veya ikiden fazla listeyi birleştirmenin birkaç yolu vardır.

* İki\_değişken\_toplam=birinci\_değişken + ikinci\_değişken

pozitif\_sayılar=[0,1,2,3,4,5,6,7,8,9]

negatif\_sayısalr=[-1,-2,-3,-4,-5,-6,-7,-8,-9]

tam\_sayılar=pozitif\_sayılar+negatif\_sayısalr

print(tam\_sayılar)#[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1, -2, -3, -4, -5, -6, -7, -8, -9]

* extend() komutu kullanarak birinci değişkene ikinci değişken eklenir.

pozitif\_sayılar=[0,1,2,3,4,5,6,7,8,9]

negatif\_sayısalr=[-1,-2,-3,-4,-5,-6,-7,-8,-9]

pozitif\_sayılar.extend(negatif\_sayısalr)

print('tamsayılar:',pozitif\_sayılar)#tamsayılar:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1, -2, -3, -4, -5, -6, -7, -8, -9]

## Python'da Liste Sayma

count() komutu ile listedeki değişkenleri sayar.

Bir dizi sayı olsun ve kaç kişi bu dizide 25 yaşında olduğunu bulalım. Örneğimizi inceleyelim 4 adet 25 yaşında kişi olduğunu program çıktısı gösteriyor.

yaş=[24,25,17,29,28,25,24,27,26,28,24,25,26,23,24,25]

print(yaş.count(25))#4

## Python'da Bir Öğenin İndeksini Bulma

index() komutu kullanılır.

meyveler  = [ 'muz' , 'portakal' , 'mango' , 'limon' ]

print ( meyveler . index ( 'portakal' )) #1

## Python'da Listeyi Ters Çevirme

Listenin sonundaki eleman başa gelir baştaki eleman ise sona gelir.reverse() komutu kullanılır.

meyveler  = [ 'muz' , 'portakal' , 'mango' , 'limon' ]

meyveler . reverse()

print(meyveler)#['limon', 'mango', 'portakal', 'muz']

## Python'da Liste Öğelerini Sıralama

* sort(): listeyi sıralar artan sıralama: lıst.sort() azalan sıralama:lıst.sort(reverse=True) şeklinde yazılarak sıralama yapılır. Aşağıdaki örneği incelersek değişkenlerin alfabetik sıralamaya göre listelediğini görmekteyiz.

meyve=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

meyve.sort()

print(meyve)#['karpuz', 'kavun', 'mango', 'muz', 'portakal']

* sorted():sıralama yapar fakat print() komutunun içine yazılır.

meyve=['muz', 'portakal', 'karpuz', 'kavun', 'mango']

print(sorted(meyve))#['karpuz', 'kavun', 'mango', 'muz', 'portakal']

Bu yazıda sizlere elimden geldiğince liste yapıları ve kullanılan genel komutlar ile ilgili bilgi vermeye çalıştım. Listeleme analiz için çok önemlidir artan azalan yapılar gibi komutlar bana İstatistik’ deki kümülatif dağılım fonksiyonu gibi konularda işe yarayacağını hatırlatıyor. Umarım beğendiğiniz ve anlaşılır bir yazı olmuştur.