Koleksiyonlar

Temel kavramlar

index(indis):

Bir koleksiyon içerisindeki bir elemanın bulunuduğu konumu temsil eden değer.

item(eleman/öğe):

Bir koleksiyonu oluşturan herbir parçaya verilen isim.

[] operatörü:

Bu operatör bir koleksiyon yapısının hemen ardından gelerek, koleksiyonun hangi elemanına ulaşacağımız bilgisini(indis veya anahtar) alır ve o elemana ulaşmamızı sağlar.

mutable(değişebilir)/immutable(değişemez):

Bazen programlama dillerindeki bazı yapılar içeriği değiştirilebilir olurken, bazı yapıların içeriği değiştirilemez olur. Örneğin bir listenin herhangi bir elemanınun değerini indis numrası ile ulaşıp değiştirebilirken bir demetin içeriğini değiştiremezsiniz.

list(liste):

- En çok kullanılan koleksiyon türüdür. liste oluşturmak için köşeli parantezler kullanılır[]
- değişebilir bir yapıya sahiptir.
- En sık kullanılan methodları
  - append() Listenin sonuna bir eleman ekler.
  - clear() Listedeki tüm öğeleri kaldırır
  - copy() Listenin bir kopyasını döndürür
  - count() Belirtilen değere sahip öğelerin sayısını döndürür
  - extend() Bir listenin öğelerini (veya yinelenebilir herhangi bir öğeyi) geçerli listenin sonuna ekler
  - index() Belirtilen değere sahip ilk elemanın indeksini döndürür
  - insert() Belirtilen konuma bir öğe ekler
  - pop() Belirtilen konumdaki elemanı kaldırır
  - remove() Belirtilen değere sahip ilk öğeyi kaldırır
  - reverse() Listenin sırasını tersine çevirir
  - sort() Listeyi sıralar

```
# %% listeler
l1 = []  # boş liste **
l2 = list()  # boş liste

ogler = ["ahmet efe", "zeynep", "reyyan", "yusuf", "mert",
         "kerem", "tarık"]
mix = [1, 2, 32.0, "hasan", True, True, False]
```

```
print(ogler)
print(mix)
print(ogler[1])
ogler[3] = "yusuf aras"
print(ogler)

for isim in ogler:
    print(isim)
```

```
['ahmet efe', 'zeynep', 'reyyan', 'yusuf', 'mert', 'kerem', 'tarık']
[1, 2, 32.0, 'hasan', True, True, False]
zeynep
['ahmet efe', 'zeynep', 'reyyan', 'yusuf aras', 'mert', 'kerem',
'tarık']
ahmet efe
zeynep
reyyan
yusuf aras
mert
kerem
tarık
```

```
# kullanıcının girdiği sayıda bir öğrenci listesi oluştur. Her
elemanaı 50-100 arası rastgele notlar ver. Sonra ortalamayı bul.

import random

# sayi = random.randint(50,100)
# print(sayi)

# kullanıcıdan öğrenci sayısı al
# boş liste tanımla
# kullanıcının girdiği değer kadar dönen bir for döngüsü başlat
# döngünün her adımında rastegele bir sayı tut ve her tuttulan sayıyı
listeye ekle.
adet=int(input("öğrenci sayısını giriniz:"))
liste=[]

for t in range(adet): # eğer döngü değişkeni gereksiz ise değişken _
yapılabilir.
    tutulan = random.randint(50,100)
    liste.append(tutulan)

print(liste)

toplam=0
for nt in liste:
    toplam+=nt

print(toplam/adet)
```

```python
r1 = range(5) # 0 1 2 3 4
r2 = range(2,6) # 2 3 4 5
r3 = range(5,11,2) # 5 7 9
r4 = range(11,2,-4) # 11 7 3
print(r1)

# for değişken in koleksiyon:
#     işlemler
for i in r2:
    print(i)

# ilkel(değer) tipler ve referans tipleri

# ilkel tipler int float complex boolean

a=5
b=a
b=8

print(a,b)

l1 = [25,30,22]
l2 = l1.copy()
l3 = l1.copy()

l2[2]=111
l1.clear()
print(l1,id(l1))
print(l2,id(l2))
print(l3,id(l3))

# append methodu: listenin sonuna eleman/item/öğe ekler
l4 = [1,2,3,4,5]
# l4 listesinin sonuna 6 elemanını ekleyelim
l4.append(6)
l4.append(3)
print("l4 e 6 ve 3  eklendi",l4)
# count methodu: listede bir elemandan kaç tane olduğunun bilgisini
verir.
say1 = l4.count(77)
say2 = l4.count(3)
print(say1)
print(say2)
# insert methodu: istenilen bir indis yerine ekleme yapar diğer
elemanlar bir sağa kayar
l4.insert(2,77)
print(l4)
# index methodu: listede bir değeri arar. Değer listede yok ise hata
verir. Var ise değerin listedeki ilk bulunduğu indis değerini verir.
indis=l4.index(3)
```

```
print("3 değerinin indisi",indis)
# remove methodu: listedeki bir değeri siler ve diğer elemanlar sola
kayar
l4.remove(5)
print(l4)
# pop methodu: listeden belirtilen indisteki değeri siler. eğer indis
verilmez ise en son değeri siler. Sildiği değeri döndürür.
d1=l4.pop() # listeden son değeri sil. silinen değeri d1 değişkenine
aktar
print(l4)
print(d1)
d2=l4.pop(2)# 2. indis değerini sil ve d2 ye o değeri aktar
print(l4)
print(d2)
# reverse methodu: listeyi ters çevirir.
l4.reverse()
print(l4)
# sort methodu: listeyi küçükten büyüğe sıralar
l4.sort()
print(l4)
```

tuple(demet):

- Birden çok öğeyi tek bir değişkende depolamak için kullanılır. Tuple, Python'da veri koleksiyonlarını depolamak için kullanılan 4 yerleşik veri türünden biridir, diğer 3'ü Liste, Küme ve Sözlük'tür ve hepsi farklı niteliklere ve kullanıma sahiptir.
- Demet, sıralı ve değiştirilemez bir koleksiyondur.
- Demetler yuvarlak parantezlerle yazılır.()
- En sık kullanılan methodları
  - count() Belirtilen değere sahip öğelerin sayısını döndürür
  - index() Belirtilen değere sahip ilk elemanın indeksini döndürür

```
# örnekler
```

Kümeler(set):

- Kümeler, birden çok öğeyi tek bir değişkende depolamak için kullanılır.
- Python'da veri koleksiyonlarını depolamak için kullanılan 4 yerleşik veri türünden biridir, diğer 3'ü List, Tuple ve Dictionary'dir ve hepsi farklı niteliklere ve kullanıma sahiptir.
- A set is a collection which is unordered, unchangeable*, and unindexed.
- Kümeler süslü parantezlerle yazılır.{}
- En sık kullanılan methodları
  - copy() Kümenin bir kopyasını döndürür
  - difference() İki veya daha fazla küme arasındaki farkı içeren bir küme döndürür
  - difference_update() Bu kümedeki, belirtilen başka bir kümede de bulunan öğeleri kaldırır.
  - discard() Belirtilen öğeyi kaldır
  - intersection() Diğer iki kümenin kesişimi olan bir küme döndürür

- intersection_update() Belirtilen diğer küme(ler)de bulunmayan öğeleri bu kümeden kaldırır.
- isdisjoint() İki kümenin kesişimi olup olmadığını döndürür
- issubset() Başka bir kümenin bu kümeyi içerip içermediğini döndürür.
- issuperset() Bu kümenin başka bir küme içerip içermediğini döndürür
- pop() Kümeden bir öğeyi kaldırır
- remove() Belirtilen öğeyi kaldırır
- symmetric_difference() İki kümenin simetrik farklarını içeren bir küme döndürür
- symmetric_difference_update() bu kümeden ve diğerinden simetrik farkları ekler
- union() Kümelerin birleşimini içeren bir küme döndürür
- update() Bu kümenin ve diğerlerinin birleşimiyle kümeyi güncelleyin

```
# örnekler
```

Sözlükler(Dictionary):

- Sözlükler, veri değerlerini anahtar:değer çiftlerinde depolamak için kullanılır.
- Sözlük, sıralı*, değişken ve tekrara izin vermeyen bir koleksiyondur.
- Sözlükler süslü parantezlerle yazılır ve anahtarları ve değerleri vardır. {k:v}
- En sık kullanılan methodları:
  - clear() Sözlüğün tüm öğelerini kaldırır
  - copy() Sözlüğün bir kopyasını döndürür
  - fromkeys() Belirtilen anahtarları ve değeri içeren bir sözlüğü döndürür
  - get() Belirtilen anahtar öğelerin değerini döndürür
  - items() Bir demet içeren bir liste döndürür her bir anahtar değer çifti için keys() Sözlüğün anahtarlarını içeren bir liste döndürür
  - pop() Belirtilen anahtara sahip öğeyi kaldırır
  - popitem() Son eklenen anahtar-değer çiftini kaldırır
  - setdefault() Belirtilen anahtarın değerini döndürür. Anahtar yoksa: belirtilen değere sahip anahtarı ekleyin
  - update() Sözlüğü belirtilen anahtar-değer çiftleri ile günceller
  - values() Sözlükteki tüm değerlerin bir listesini döndürür

```
#örnekler
```