1
1

PYTHON PROGRAMLAMA-201 EĞİTİMİ **ANIL NEBİ ŞENTÜRK**

EĞİTMEN:M.VAHİT KESKİN **#turkcellgeleceğiyazanlar 15.10.2020**

1)*LİSTE OLUŞTURMA:* listeler;değiştirliebilirler,kapsayıcıdır(farklı tipte veriler tutabilirler),sıraldır.

- ★ *Liste oluşturmanın iki farklı yolu vardır:1)[];köşeli parantez kullanımı.Örneğin bir sınıfa ait popülsayonda o sınıfda ki bazı öğrencilerin notlarını örneklem olarak seçebiliriz.*
- ➢ *Notlar=[90,80,70,50] ,type(notlar):out="list"*
- ○ *liste=["a",90.3,56]*
- ○ list_geniş=["a",90.3,56,notlar]
- ○ len(list_geniş)=4

2)liste içi tip sorgulama: tum_liste=[liste,list_geniş]

- list_geniş[0]="a"
- type=(liste[2])=56

3)LİSTELERE ELEMAN EKLEME&DEĞİŞTİRME VE SİLME İŞLEMLERİ: Eleman değiştirme=Örneğin liste=["Ali","Veli","Berke","Aysel"] liste[1]= "veli" liste[1]="velinin babası" = Liste[1]:velinin babası

Liste[0;3] oıncı elemandan üçünce elemana kadar:"ali","velinin babası","berke"

Liste[0;3]="alinin babası","velinin babası","berkenin babası"

Eleman ekleme: Liste adında yeni bir değişken atanır.

Liste=liste+[“Kemal”]

Eleman Silme:**”delete”** komutu del liste[2] yeni listede “Berke”olmayacaktır.

- ★ ***Dir(liste)*** komutu ile liste ile ilgili yapabileceğimiz tüm methodları console ekranında görüntüleyebiliriz.

Append ve Remove methodu: liste.append(“berkcan”):listenin sonuna daha önceden çıkarmış olduğumuz “berkcan” stringini ekleyebiliriz.

Liste.remove(“berkcan”):append methodu ile eklemiş olduğumuz “berkcanı”remove komutu ile tekrardan çıkarabiliriz.

Indexe göre “insert ve pop”: liste.insert(0,”anıl”): listenin o ıncı indexine anıl stringini ekle.

Liste.insert(5,”berk”):5.index olarak Berk’i ekler.

Liste.pop(0):0 ıncı indexi yok eder.

#count

liste=("ali","veli","ışık","ali","veli")

liste.count("ali")

#coppy

liste_yedek=liste.coppy()

#extend: liste.extend([“a”,”b”,10]) :listeye yeni listeyi ekler.

#index: liste.index(“ali”):0

#reverse: elemanları ters çevirme işlemidir. Liste.reverse()

#sort : elemanları sıralamak için kullanılır liste.sort() numeric elemanları sıralar.

#clear:elemanları temizlemek için kullanılır.liste.clear()

TUPLE OLUŞTURMA : tuple bir veri yapısıdır.**”listeden farkı değiştirilemez olmasıdır.” Aynı zamanda sıralı ve kapsayıcıdır.**

t=("ali","veli",1,2,3.2,[1,2,3,5])

t=("eleman")

Tuple eleman işlemleri: t=[0]=”ali t=[0,3] =”ali”,”veli”,1

SÖZLÜK(dictionary) OLUŞTURMA : 1)kapsayıcıdır

2)sıraszıdır

3)değiştirilebilir

```
sozluk={"REGRESYON":"Model",
        "LOJ":"lojmean",
        "CART":"Classification"}
```

“regresyon,loj,cart”:key

“model,lojmean,classification”:dictionary

Len(sozluk)=3 sadece key olanları sayıyoruz.

Sözlük Eleman seçme işlemleri: sözlük[loj]:”lojeman”,

SET(KÜME)OLUŞTURMA: s=set()

Set eleman ekleme çıkarma: l=["geleceği_yazanlar"]

```
l=["geleceği_yazanlar"]
s=set(l)
s.add("ile")
s.remove("ali")
```

s.discard("ali")

*Differences and symetric differences:#difference() ile iki kümenin farkını ya da **"-"**ifadesini*

***"intersection()"** iki küme kesişimini ya da "&"*

*"**union()"** iki kümenin birleşimi*

***"symetric_ difference()"** ikisinde de olmayanlar*

set1=([1,3,5])

set2=([1,2,3])

set1.difference(set2)

Set1.symetric_differences(set2)

set.intersection(set2)

set1.union(set2)

set1.intersection_update(set2)

SETLERDE QUERY(SORGU)İŞLEMLERİ: set1=set([7,8,9])

set2=set([5,6,7,8,9,10]) iki kümenin kesişiminin boş olup olmayacağını sorgulayacağız.

Bir kümenin bütün elemanlarının başka bir kümede yer alıp almadığını sorguluyalım.

set1.issubset(set2)

Out[102]: True

Bir kümenin bir diğer kümeyi kapsaması

set2.issuperset(set1)

Out[103]: True