SETLER(SETS)

Set belirli bir diziden oluşmaksızın her tip değişkeni tanımlayabileceğimiz bir örneklem olarak düşünülebilir. Python'da set, birbirine benzemeyen ögeleri saklamak için kullanılır ve bir küme olarak düşünülebilir kümeler arasında birleşim, kesişim, fark, alt küme bulmak mümkündür.

# Set Oluşturma

Bir küme olduğunu düşünelim demiştik oluşturmak için küme parantezleri, {} veya yerleşik set() işlevi kullanırız. Aşağıda örnek de iki şekildeki tanımı görebiliriz.

st{}

ve ya

st=set()

Örnek bir set oluşturalım

st{'istanbul','eskişehir','bursa','ankara'}

# Setin Uzunluğunu Alma

Bir dizi uzunluğu bulmak istiyorsak len() komutu kullanırız.

Örnek:1

st={'istanbul','eskişehir','bursa','ankara'}

len(st)#4

Örnek:2

meyve={'ananas','karpuz','muz','çilek','elma'}

uzunluk=len(meyve)

print(uzunluk)

## 5

# Bir Öğeyi Var Olup/Olmadığı Kontrol Etme

“memebership operatör “olarak adlandırılan bu komutta içerisinde olup olmadığını merak ettiğim ögeyi bulmak için yazdığım modülü inceleyelim.

ÖRNEK:1

st={'istanbul','eskişehir','bursa','ankara'}

print('giitiği yerler arasında istanbul var mı?','istanbul' in st)#giitiği yerler arasında istanbul var mı? True

ÖRNEK:2

meyve={'ananas','karpuz','muz','çilek','elma'}

print('çilek'in meyve)#True

# Bir Sete Öğe Ekleme

Bir set oluşturulduktan sonra herhangi bir öğeyi değiştiremeyiz fakat ek öğeler de ekleyebiliriz.Aşağıdaki örnek de add() komutu kullanılarak yeni öge eklenmesini görmekteyiz.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

meyve.add('kivi')

print(meyve)#{'çilek', 'karpuz', 'ananas', 'elma', 'kivi', 'muz'}

* Birden fazla öge ekleme işlemini aşağıdaki örnekte update() komutu ile yapıldığını görmekteyiz.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

meyve.update(['kivi','nar','erik'])

print(meyve)# {'erik', 'nar', 'çilek', 'karpuz', 'ananas', 'elma', 'kivi', 'muz'}

# Bir Setten Öğe Çıkarma

Remove() komutunu kullanarak bir kümeden bir öğeyi kaldırabiliriz . Bazen bulunmayan bir elemanı çıkarabilirim dolayısıyla olup olmadığını önce kontrol etmem gerekir. Ancak, discard() yöntemi herhangi bir kontrol gerek duymadan istenen öğeyi çıkartır.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

meyve.remove('ananas')

print(meyve) # {'çilek', 'karpuz', 'elma', 'muz'}

* pop() komutu kullanarak rasgele bir ögeyi kaldırır kaldırılan ögenin ne olduğu ile ilgilenirsek aşağıdaki örnekteki gibi bulabiliriz.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

kayıp\_meyve=meyve.pop()

print(kayıp\_meyve) # çilek

# Bir Seti Silme

Kümenin kendisini silmek istiyorsak del operatörünü kullanırız.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

del meyve

print(meyve) # error

# Listeyi Sete Dönüştürme

Bir küme oluşturmak için küme parantezleri, {} veya yerleşik set() işlevi kullanırız.

Örnek:1

st{}

ve ya

st=set()

örnek:2

meyve=('ananas','karpuz','muz','çilek','elma')

meyve=set(meyve)

print((meyve))  # {'karpuz', 'çilek', 'ananas', 'elma', 'muz'}

# Birleştirme Setleri

union() veya update() yöntemini kullanarak iki kümeyi birleştirebiliriz .

* union() komutunu kullandığımız bu örneği inceleyeim.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

kuru\_yemiş={'fındık','fıstık','leblebi'}

meyve.union(kuru\_yemiş)

## {'ananas', 'elma', 'fındık', 'fıstık', 'karpuz', 'leblebi', 'muz', 'çilek'}

* update() komutu belirli kümeyi belirli kümeye sırasıyla ekler yani önce hangisini yazarsak onun elemanları ilk sıralarda bulunur. Ben ‘kuru\_yemiş’ değişkenini önce ‘meyve’ değişkenini sonra eklemek istedim aşağıdaki şekilde komutu yazdım.

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

kuru\_yemiş={'fındık','fıstık','leblebi'}

kuru\_yemiş.update(meyve)

print(kuru\_yemiş)

#{'fındık', 'fıstık', 'leblebi', 'karpuz', 'çilek', 'ananas', 'elma', 'muz'

# Kesişim Ögelerini Bulma

Her iki kümede de bulunan bir dizi öğe döndürür intersection() komutu kullanılır. Örneğe bakalım

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

kuru\_yemiş={'fındık','fıstık','leblebi','elma','armut'}

meyve.intersection(kuru\_yemiş)

#{'elma'}

# Alt Kümeyi ve Süper Kümeyi Kontrol Etme

Bir küme, diğer kümelerin bir alt kümesi veya bir üst kümesi olabilir:

Alt küme: issubset()

Üst küme: issuperset

Örnek:

meyve={'ananas','karpuz','muz','çilek','elma'}

kuru\_yemiş={'fındık','fıstık','leblebi','elma','armut'}

alt\_küme\_1={'ananas','küpe'}

alt\_küme\_2={'çilek','karpuz','elma'}

alt\_küme\_2.issubset(meyve) # True

alt\_küme\_1.issubset(meyve) #False

meyve.issuperset(alt\_küme\_2)# True

# İki Set Arasındaki Farkı Kontrol Etme

İki küme arasındaki farkı bize getirir yani çıkarma işlemi gibi düşünülebilir. difference() komutu kullanılır dilerseniz örneğimizde inceleyelim.

Örnek:

tabak\_1={'elma','kivi','muz','şeftali'}

tabak\_2={'elma','kivi','portakal'}

tabak\_1.difference(tabak\_2)  # tabak 1 de kalananın çıktısını verir.

## {'muz', 'şeftali'}

tabak\_2.difference(tabak\_1) # tabak 2 de kalanın çıktısını verir.

# portakal

# İki Küme Arasındaki Simetrik Farkı Bulma

İki küme arasındaki simetrik farkı döndürür. Bu, matematiksel olarak her iki kümede de bulunan öğeler dışında her iki kümedeki tüm öğeleri içeren bir küme döndürdüğü anlamına gelir((A\B) ∪ (B\A)) symmetric\_difference() komutu ile çalışır aşağıdaki örnekte inceleyelim.

Örnek:

tabak\_1={'elma','kivi','muz','şeftali'}

tabak\_2={'elma','kivi','portakal'}

tabak\_1.symmetric\_difference(tabak\_2)   # ikisinde de ortak olmayanı verir.

#{'muz', 'portakal', 'şeftali'}

# Birleştirme Setleri

İki kümenin ortak bir öğesi veya öğeleri yoksa, bunlara ayrık kümeler diyoruz. isdisjoint() komutu ile kullanarak iki kümenin ortak mı yoksa ayrık mı olduğunu kontrol edebiliriz. Örneğimizi inceleyelim.

Örnek:1

tam\_sayılar={-1,-2,-3,-4,-5,0,1,2,3,4,5,6,7,8,9}

doğal\_sayılar={0,1,2,3,4,5,6,7,8,9}

ondalık\_sayılar={10.25,11.45,14.2}

tabak\_1={'elma','kivi','muz','şeftali'}

doğal\_sayılar.isdisjoint(ondalık\_sayılar)

#True ortak öğe yok

tabak\_1.isdisjoint(doğal\_sayılar)

# True ortak öğe yok

doğal\_sayılar.isdisjoint(tam\_sayılar)

# False ortak öğe var {0,1,2,3,4,5,6,7,8,9}