

```
In [1]: liste=['Ali','Ahmet','Kemal','Ayşe','Hakan'] # list (liste)
liste
```

```
Out[1]: ['Ali', 'Ahmet', 'Kemal', 'Ayşe', 'Hakan']
```

```
In [2]: type(liste)
```

```
Out[2]: list
```

```
In [3]: # set (küme) aynı eleman sadece 1 kere tanımlanabilir
kume={'Ali','Ahmet','Kemal','Ayşe','Hakan'}
kume
```

```
Out[3]: {'Ahmet', 'Ali', 'Ayşe', 'Hakan', 'Kemal'}
```

```
In [4]: type(kume)
```

```
Out[4]: set
```

```
In [5]: # demet (tuple) birden fazla eleman tutabilen veri tipidir
# tuple (demet) bir kere oluşturulunca değiştirilemez
demet=('Ali','Ahmet','Kemal','Ayşe','Hakan')
demet
```

```
Out[5]: ('Ali', 'Ahmet', 'Kemal', 'Ayşe', 'Hakan')
```

```
In [6]: type(demet)
```

```
Out[6]: tuple
```

```
In [7]: # enumerate() fonksiyonu verileri numaralandırır. for döngüsü ile kullanılabilir
list(enumerate('Python'))
```

```
Out[7]: [(0, 'P'), (1, 'y'), (2, 't'), (3, 'h'), (4, 'o'), (5, 'n')]
```

```
In [8]: gun=['Pazartesi','Salı','Çarşamba','Perşembe','Cuma','Cumartesi','Pazar']
# for döngüsü ile enumerate() fonk. döndürdüğü indeks ve değeri ayrıştırmak
for i , deger in enumerate(gun,start=1):
    print(f"{i}. gün: {deger}")
```

```
1. gün: Pazartesi
2. gün: Salı
3. gün: Çarşamba
4. gün: Perşembe
5. gün: Cuma
6. gün: Cumartesi
7. gün: Pazar
```

```
In [9]: # for döngüsü ile enumerate() fonk. döndürdüğü indeks ve değeri ayrıştırmamak
for x in enumerate(gun,start=1):
    print(x)
```

```
(1, 'Pazartesi')
(2, 'Salı')
(3, 'Çarşamba')
(4, 'Perşembe')
(5, 'Cuma')
(6, 'Cumartesi')
(7, 'Pazar')
```

```
In [10]: # küme (set) sırasız tekil eleman topluluğudur
# küme elemanlarına erişim indeks ile olmaz
# kümeye elemanları doğrudan değiştiremeyen türler (tuple, str, int, float) ekl
kume1={'elma', 'nar', 'muz', 'erik', 'armut'} # string elemanlar
kume2={1,3,10,2,7} # tamsayı elemanlar
kume3=set('adana')
```

```
In [11]: kume1
```

```
Out[11]: {'armut', 'elma', 'erik', 'muz', 'nar'}
```

```
In [12]: kume2
```

```
Out[12]: {1, 2, 3, 7, 10}
```

```
In [13]: kume3
```

```
Out[13]: {'a', 'd', 'n'}
```

```
In [14]: bos_kume=set() # boş küme içinde eleman olmayan küme
bos_kume
```

```
Out[14]: set()
```

```
In [15]: kume3.clear() # küme içindeki elemanları siler
kume3
```

```
Out[15]: set()
```

```
In [16]: A={0,2,4,6,8}
B={1,2,3,4,5}
```

```
Out[16]: ({0, 2, 4, 6, 8}, {1, 2, 3, 4, 5})
```

```
In [17]: A.difference(B) # A fark B kümesi
```

```
Out[17]: {0, 6, 8}
```

```
In [18]: B.difference(A) # B fark A kümesi
```

```
Out[18]: {1, 3, 5}
```

```
In [19]: A.intersection(B) # A kesişim B
```

```
Out[19]: {2, 4}
```

```
In [20]: A.union(B) # A birleşim B
```

```
Out[20]: {0, 1, 2, 3, 4, 5, 6, 8}
```

```
In [21]: A.issubset(B) # A, B'nin alt kümesi mi?
```

```
Out[21]: False
```

```
In [22]: B.issubset(A) # B, A'nın alt kümesi mi?
```

```
Out[22]: False
```

```
In [23]: A={0,2,4,6,8}
         B={2,4}
         A,B
```

```
Out[23]: ({0, 2, 4, 6, 8}, {2, 4})
```

```
In [24]: A.issubset(B) # A, B'nin alt kümesi mi?
```

```
Out[24]: False
```

```
In [25]: B.issubset(A) # B, A'nın alt kümesi mi?
```

```
Out[25]: True
```

```
In [26]: A.issuperset(B) # A kapsar B
```

```
Out[26]: True
```

```
In [27]: B.issuperset(A) # B kapsar A
```

```
Out[27]: False
```

```
In [28]: A={0,2,4,6,8}
         B={0,2,4,6,8}
         A,B
```

```
Out[28]: ({0, 2, 4, 6, 8}, {0, 2, 4, 6, 8})
```

```
In [29]: A.issuperset(B) # A kapsar B
```

```
Out[29]: True
```

```
In [30]: B.issuperset(A) # B kapsar A
```

```
Out[30]: True
```

```
In [31]: A={0,2,4,6,8}
         B={2,4}
         A,B
```

```
Out[31]: ({0, 2, 4, 6, 8}, {2, 4})
```

```
In [32]: x=2
         x in A # x(2) A'nın içinde mi?
```

```
Out[32]: True
```

```
In [33]: x in B
```

```
Out[33]: True
```

```
In [34]: y=6
         y in B # y(6) B'nin içinde mi?
```

```
Out[34]: False
```

```
In [35]: x not in A # x(2) A'nın içinde değil mi?
```

```
Out[35]: False
```

```
In [36]: y not in B # y(6) B'nin içinde değil mi?
```

```
Out[36]: True
```