

# PYTHON



## İÇERİK

- ✓ **Demetler**
- ✓ **Sözlükler**

## KAZANIMLAR

- ✓ Python'da demet fonksiyonu ve kullanım alanlarını öğrenir.
- ✓ Python'da sözlüklerin ne olduğunu öğrenir.
- ✓ Sözlüklerin söz dizimini ve kullanım alanlarını öğrenir



## Demetler

Python'da demetler diye bir değişken tipimiz var. Demetler listelere , çok benziyor. Fakat iki temel farkı var. Bunlardan birisi köşeli parantez yerine normal parantez kullanmamız ikincisi ise listenin eleman sayısının değişmemesi. Yani demetleri elemanları değiştirilemeyen listeler gibi düşünebiliriz. Demetlerde elemanları sadece okuyabiliyoruz. Herhangi bir ekleme çıkarma silme yapamıyoruz.

Demetleri nerede kullanacağız?

Örneğin programımızda Türkiye'deki şehirlerin plaka numaralarını kullanacağız. Elimizde değişmeyecek bir liste var. Herhangi bir ekleme çıkarma silme yapmayacağız. İşte burada liste yerine demet kullanabiliyoruz.

Demetlerin özellikleri

- Demetler öğeleri sıralıdır, değiştirilemez ve yinelenen değerlere izin verir.
- Demetler öğeleri indekslenir, ilk öğenin indeksi vardır `[0]`, ikinci öğenin indeksi vardır `[1]` vb.
- Demetler dizine eklendiğinden, demetler aynı değere sahip öğelere sahip olabilir

Demetler ile ilgili ilk örneğimiz ile başlayalım...

Türkiye'deki beş şehri yazalım

```
sehirler = ("İstanbul", "Adana", "Bitlis", "Kayseri", "Malatya")  
  
print(sehirler)
```

Bu örneğimiz ile sehirler isminde bir demet tanımladık. Bu demet içerisine hiçbir zaman değişmeyecek olan şehir isimlerimizi ekledik.

Şimdi bu dizi içerisine listelerdeki gibi eleman ekleyip ekleyemeyeceğimizi kontrol edelim.

Demetimizde bulunan ilk elemanı değiştirmeyi deneyelim



```
sehirler = ("İstnabul", "Adana", "Bitlis", "Kayseri", "Malatya")  
  
sehirler[0]="Antalya"  
  
print(sehirler)
```

Bu kod satırını yazdığımız takdirde aşağıdaki gibi bir hata ile karşılaşacağız çünkü demetlerde bulunan elmenler değişmez.

```
===== RESTART: C:/Users/Bahar/AppData/Local/Programs/Python/Python39/wd.py =====  
Traceback (most recent call last):  
  File "C:/Users/Bahar/AppData/Local/Programs/Python/Python39/wd.py", line 2, in  
    <module>  
      sehirler[0]="Antalya"  
TypeError: 'tuple' object does not support item assignment  
>>> |
```

Demetteki öge sayısını yazdırmak için len() fonksiyonu kullanılır.

```
sehirler = ("İstnabul", "Adana", "Bitlis", "Kayseri", "Malatya")  
  
print(len(sehirler))
```

Demetler farklı veri türleri ile oluşturulabilir.

```
demet1 = ("elma", "muz", "kiraz")  
  
demet2 = (1, 5, 7, 9, 3)  
  
demet3 = (True, False, False)  
  
print(demet1)  
  
print(demet2)  
  
print(demet3)
```



**Not:** Bir demet içerisinde farklı ver tipleri bulunabilir. Örneğin;

```
Demet1 = ("abc", 34, True, 40, "kadın")
```

Köşeli parantezlerin içindeki dizin numarasına bakarak demet öğelerine erişebilir ve yazdırabiliriz.

```
meyve = ("elma", "muz", "kiraz")  
print(meyve[1])
```

Birden fazla öğeyi yazdırmak için sırasıyla indis numaralı yazılı köşeli parantes içine araya iki nokta (:)

Konulur.Örnek print(meyve[1:2]) böylecek demet içersindeki iki farklı eleman yazdırılabilir.

### Sıra Sizde

**plakalar** demetinde bulunan ilk öğeyi yazdırmak için boşluğu uygun şekilde doldurun .

```
plakalar = ("34", "44", "18")  
  
print( )
```

**Cevap:** plakalar[0]

plakalar demetinde ki öge sayısını yazdırmak için boşluğu uygun şekilde doldurun

```
plakalar = ("34", "44", "18")  
  
print( )
```



## Sözlükler

Şu ana kadar Python programlama dilinde veri tipi olarak karakter dizilerini, sayıları, listeleri, demetleri öğrendik. Bu bölümde yine Python'daki önemli veri tiplerinden birini inceleyeceğiz. Bu defa inceleyeceğimiz veri tipinin adı sözlük. İngilizcede karşılığı dictionarydir.

Öteki veri tiplerinde olduğu gibi, sözlüklerin de birtakım metotları vardır. İşte bu bölümde hem genel olarak sözlüklerden söz edeceğiz, hem de bu veri tipinin metotlarını inceleyeceğiz.

Python programlama dilindeki sözlük veri tipi, gerçek hayatta 'sözlük' denince aklınıza gelen şeye çok benzer. Mesela gerçek hayatta 'kitap' kelimesinin İngilizce bir sözlükteki karşılığı book kelimesidir. Dolayısıyla 'kitap' ve 'book' kelimeleri arasındaki ilişkiyi herhalde şu şekilde temsil edebiliriz:

kitap: book

Bu manzara bize 'kitap' kelimesinin karşılığının 'book' olduğunu açık bir şekilde gösteriyor. Eğer bu durumu Python'daki sözlük veri tipiyle göstermek isteseydik şöyle bir şey yazacaktık:

```
kelimeler = {"kitap": "book"}  
print(kelimeler)
```

Yukarıdaki sözlüğü nasıl tanımladığımıza çok dikkat edin. Nasıl ki listelerin ayırt edici özelliği köşeli parantezlerdi, sözlüklerin ayırt edici özelliği de küme parantezleridir.

En basit görünümü ile sözlük = {} şeklindedir.

Beraber bir telefon rehberi oluşturalım

Öncelikle isimler ve telefon numaralarından oluşan, sözlük veri tipinde bir telefon rehberi oluşturalım:

```
telefon_rehberi = {"bahar kayar" : "0532 532 32 32",  
                  "elif nur": "0543 543 42 42",  
                  "ayşe toktaş" : "0533 533 33 33",  
                  "ali kartal" : "0212 212 12 12"}
```

Daha sonra kullanıcıdan, telefon numarasını öğrenmek için bir kişi adı girmesini istiyoruz. Bunu da şu kodlar yardımıyla yapıyoruz:

```
kişi = input("Telefon numarasını öğrenmek için bir kişi adı girin: ")
```



Şimdi telefon rehberinde sorgulama yapacak olan kullanıcıya göstereceğimiz cevap için kodumuzu yazıyoruz:

```
cevap = "{} adlı kişinin telefon numarası: {}"  
print(cevap.format(kişi, telefon_rehberi[kiş]))
```

Kodumuzun örnek çıktısı aşağıdaki gibi olacaktır.

```
===== RESTART: C:/Users/Bahar/Desktop/kjkjk.py =====  
Telefon numarasını öğrenmek için bir kişi adı girin: bahar kayar  
bahar kayar adlı kişinin telefon numarası: 0532 532 32 32  
>>> |
```

Sözlükleri ilerleyen aşamalarda çok fazla kullanacağız şimdilik burada veri tiplerine ara veriyoruz ve yeni bir konumuza geçiriyoruz.