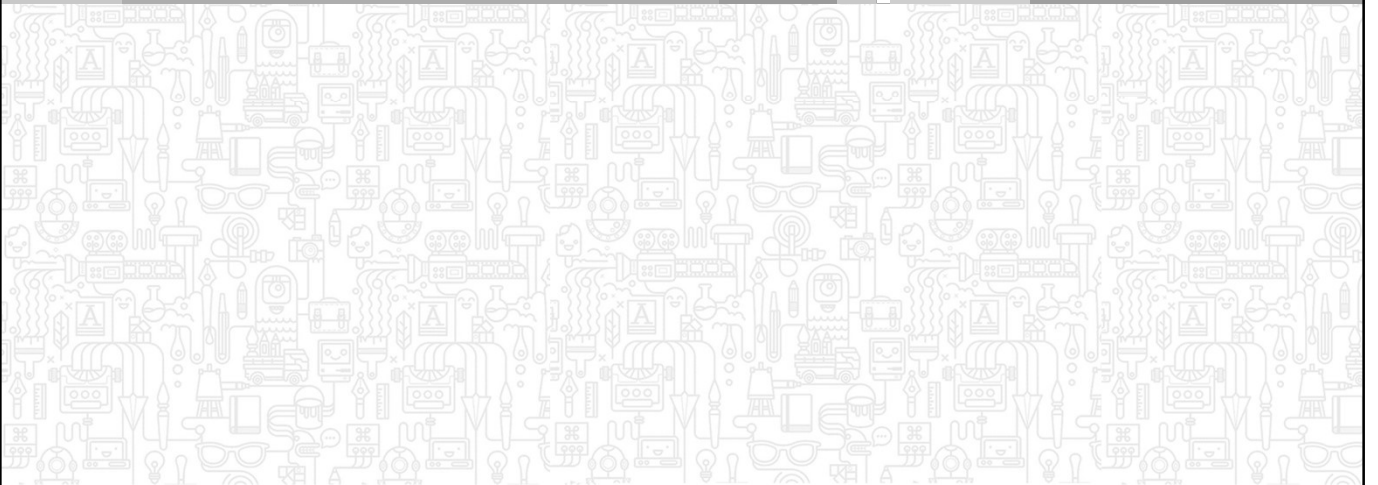


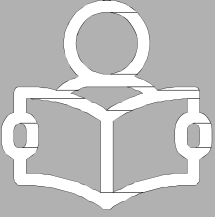
PYTHON



**Sözlükler ve Demetler**



## Kapsam



### Sözlük Oluşturma

Sözlük Anahtar ve Değerlerine Erişim

Sözlüklerde Eleman Seçme, Silme, Ekleme, Değiştirme

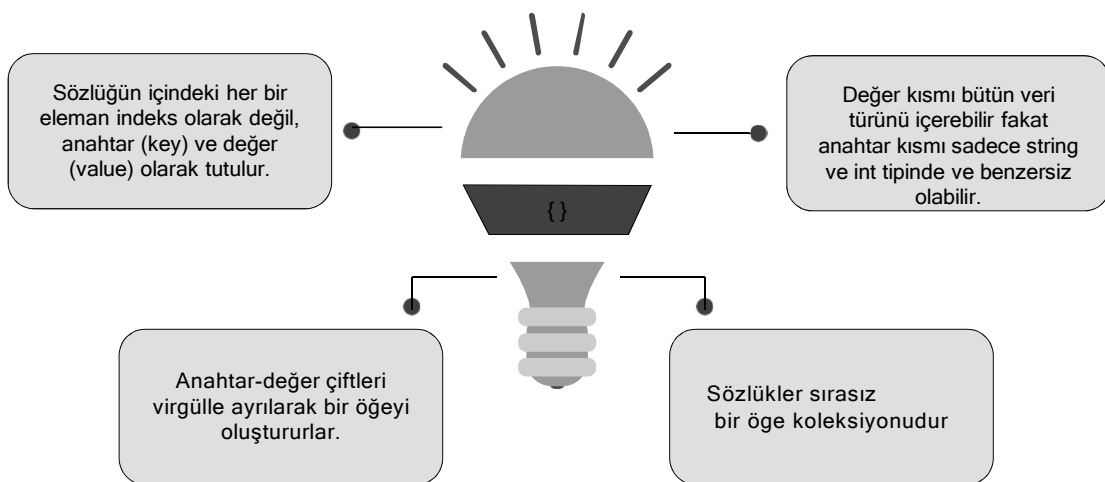
Sözlükler Üzerinde Gezinme

Demet (tuple) Oluşturma ve Eleman İşlemleri

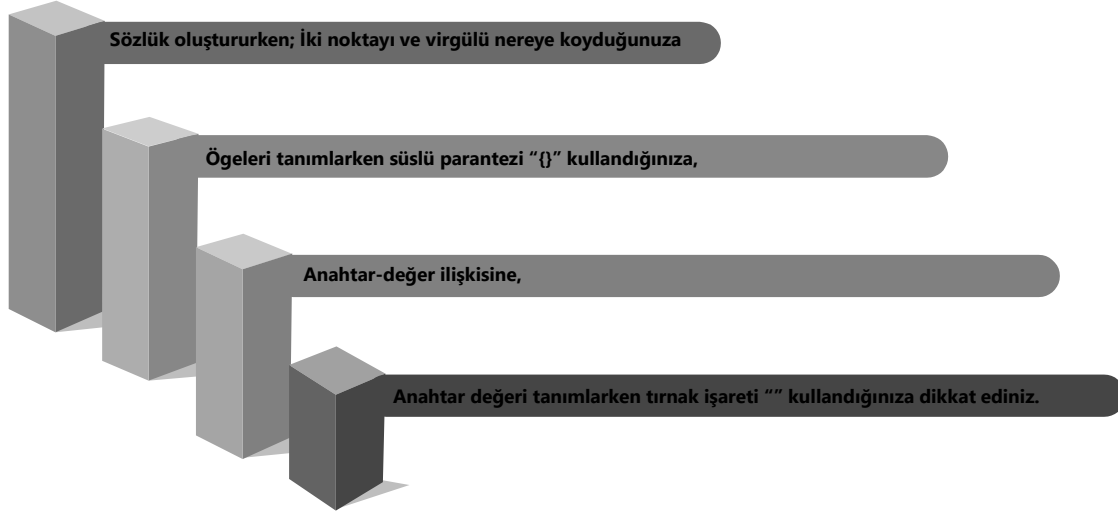
Demetlerin Temel Metotları

Demetler Üzerinde Gezinme

## Sözlükler



## Sözlük Oluşturma



```
kisisel_bilgiler={"adı":"ali","soyadı":"can","e_posta":"ali@abc.com","ülke":"Türkiye","il":"Ankara","ilçe":"Çankaya"}
print(kisisel_bilgiler)
```

## Sözlük Oluşturma

Anahtar tiplerine göre sözlükler oluşturalım.  
Sözlük oluşturmak için; { } ve dict() kullanılır.

Kod

```
sozluk1 = {} #bos bir sözlük, veya
sozluk1=dict()
print(sozluk1)
sozluk2= {1: 'adana', 2: 'adıyaman'} #int veri tipi oluşturulan anahtarlar
print(sozluk2)
sozluk3 = {'isim': 'ali', 1: [5, 4, 3]} #str ve int veri tipi ile oluşan anahtarlar
print(sozluk3)
sozluk4 = dict({1:'erik', 2:'ayva'}) #dict fonksiyonu ile
print(sozluk4)
```

## Sözlük Anahtar ve Değerlerine Erişim

Sözlük anahtar ve değerlerini listelemek için keys ve values komutları kullanılır.  
keys( ) metodu sözlükteki anahtarları yazdırır.  
values()metodu sözlükteki değerleri bize yazdırır.

Kod

```
iller={ "konya" : "42", "istanbul" : "34", "ankara" : "06" }  
print(iller.keys()) #dict_keys(['konya', 'istanbul', 'ankara'])  
print(iller.values()) #dict_values(['42', '34', '06'])
```

## Sözlükte Eleman Seçme İşlemleri

Örnek'te meyveler adında sözlük oluşturulmuştur. Bu sözlükte adı, türü ve kg adında 3 adet anahtar ve bu anahtarlara ait değerleri mevcuttur.  
meyveler sözlüğünde adı ve kg anahtarları ekrana yazdırılarak eleman seçme işlemi yapılmıştır.

Kod

```
meyveler={"Adı": "Portakal", "Türü": "Turunçgiller", "Kg": 20}  
print(meyveler["Adı"])  
print(meyveler["Kg"])
```

## Sözlükte Eleman Silme İşlemleri

pop( ) metodu, öğeyi belirtilen anahtar adıyla kaldırmak için kullanılır.  
 popitem( ) metodu, eklenen son öğeyi kaldırmak için kullanılır.  
 del, belirtilen anahtar adına sahip öğeyi veya sözlüğü tamamen silmek için kullanılmaktadır.  
 clear( ), sözlüğü boşaltmak için kullanılır, sözlük hala tanımlıdır.

Kod

```
kisi_bilgileri= {"Adı": "Kemal", "Soyadı": "Yıldırım", "Yaşı": 40, "Mesleği": "Ahçı"}
kisi_bilgileri.pop("Yaşı")
kisi_bilgileri.popitem()
del kisi_bilgileri["Adı"]
kisi_bilgileri.clear() #Sözlüğün sadece içeriğini boşaltmak için
del kisi_bilgileri #Sözlüğü tamamen silmek için
```

Örneği incelediğimizde Yaşı anahtarı pop metodu ile kaldırılmıştır. kisi\_bilgileri sözlüğü listelendiğinde sadece Adı, Soyadı ve Mesleği anahtarlarının olduğu görülür.  
 popitem metodu ile son öğe olan Mesleği kaldırılmıştır, geriye Adı ve Soyadı anahtarları kalmıştır.  
 del ile Adı silinmiş, Soyadı kalmıştır.  
 clear metodu ile sözlüğün içeriği boşaltılmış, ve son olarak del ile sözlük değişken listesinden de silinmiştir.

## Sözlüğe Eleman Ekleme İşlemleri

Sözlüklere veri ekleme işlemi yapılabilmektedir.  
 sözlükadi["anahtar"]="değer" şeklinde olmaktadır.

Kod

```
sozluk={"adi":"sami"}
sozluk['soyadi']='yilmaz'
sozluk['yasi']=40
print(sozluk)
```

## Sözlükte Eleman Değiştirme İşlemleri

Sözlükte eleman değiştirmek için anahtar kullanarak güncelleme işlemi yapılır. sözlük adı[anahtar adı]=yeni değer şeklinde yapılır.

Kod

```
sozluk = {'isim':'ahmet','yas':40}
sozluk['yas'] = 45
print(sozluk['yas'])
```

## Sözlükler Üzerinde Gezinme

Python'da sözlükler öğelerine erişmek için anahtarlarını (key) kullanılır ve değerler bu indeksler üzerinden gösterilir. Anahtarlar köşeli parantez içerisinde ya da get( ) fonksiyonu ile kullanılır. get( ) fonksiyonunda anahtar bulunamazsa KeyError yerine "None" döndürür.

Kod

```
sozluk = {'ad': 'ali', 1: [5, 4]}
print(sozluk['ad'])
print(sozluk.get(1))
```

Örnek'te sozluk adlı sözlükte ad ve 1 adında anahtarlar ve bu anahtarlara bağlı ali ve 5,4 değerleri bulunmaktadır. Ad ve 1 anahtarlarını kullanarak değerleri ekranda listelenmiştir.

## Demet (tuple) Oluşturma ve Eleman İşlemleri

Demetler veri ekleme ve çıkarmanın yapılamadığı veri yapılarıdır. Bu değiştirilemez özellikleri dışında listelere benzemektedirler. Demetler ekleme çıkarma gibi işlemlerle uğraşmadığı için daha hızlı çalışır. Eklenen verilerin program boyunca değiştirilmesinin istenmediği durumlarda kullanılmaktadır.

Kod

```
demet=()
print(demet)
```

Örnek'te boş bir demet oluşturularak, ekrana listelenmiştir.

## İndis içeren demet oluşturma

4 ögesi bulunan bir demet oluşturularak ekrana yazdırılmıştır.

Kod

```
demet = ("Python","Java",2020,"JavaScript")
print(demet)
```

Aynı zamanda bir demet tuple( ) fonksiyonu ile de oluşturulabilir.

Kod

```
demet = tuple(("Ankara","İstanbul","Kayseri"))
print(demet)
```

## Tek elemanlı demet oluşturma

Tek nesneli bir demet oluşturmak için nesneden sonra virgül konulması gerekmektedir.  
Eğer virgül koymazsanız bu değişken tipi demet değil string olarak sistemde yer alacaktır.

Kod

```
meyveler = ("erik",)  
print(meyveler)
```

## Demetlerde indisler

Demetler de sözlükler gibi indis olarak sıfırdan başlar.  
Bir indise ulaşmak için indis numarası kapalı parantez arasında girilerek yazılır.  
print(demet[-1]) demet'in en son elemanı bize vermektedir.  
print(demet[2:])'da 2 numaralı indis dahil iki numaradan sonraki tüm indislere ait elemanları listelemek için kullanılır.

Kod

```
demet = (5,10,15,20,25,30)  
print(demet[0])  
print(demet[2])  
print(demet[-1])  
print(demet[2:])
```



## Demetlerde birleştirme

Demetlerde birleştirme + ile yapılır.

Kod

```
sayılar = (0,1,2,3,4,5,6,7,8,9,) #bir demet tanımladım
harfler = ("a","b","c","d","e") #ikinci bir demet tanımladım
yeni_demet = sayılar + harfler #tanımladığım demetleri topladım
print(yeni_demet) #yeni demeti ekrana yazdırdım
```

## Demetlerde arama

if koşul yapısı ile demetlerin içerisinde aranılan elemanların olup olmadığını kontrol edilebilir.

Kod

```
harfler = ("a","b","c","d","e")
if "a" in harfler: #True çıktısı verir
    print(True)
else:
    print(False)
```

## Dilimleme ve iç içe geçirme

Demet veri yapılarında elemanlar iki parantez arasına yazılır lakin bu parantez zorunlu değildir. Yine aynı şekilde liste veri yapılarında olduğu gibi demetler de dilimlenebilir (sliceable) ve iç içe geçirilebilir (nestable) yapılardır.

Kod

```
d=((1,2),(3,4),(5,6),(7,8))
print(d[:2]) #((1, 2), (3, 4))
print(d[2:3]) #((5, 6),)
print(d[2][1])
print(d[3][0])
```

Örnek'te iç içe demetlere ait uygulamada listeleme işlemi yapılmıştır. d adında demet oluşturulmuş ve iç içe demet yapısı olarak 2. indis numarasındaki ait 1. indis elemanına ve 3. indis numarasındaki 0. indis numarasına ait elemana ulaşılmıştır.

## Demetlerin Temel Metotları

index metodu demette yer alan ögenin sırasını bulmaya yardımcı olur, öge birden fazla tekrar ediyorsa ilk sıradakini verir.  
count metodu tuple içinde aynı elemandan kaç adet olduğunu bulunmasına yardımcı olur.

Kod

```
demet = ("lale","kardelen","zambak","papatya","lale","kardelen","zambak","lale")
print(demet.index("zambak"))
print(demet.count("lale"))
```

## Demete dönüşüm

Liste tipindeki veriyi tuple'a dönüştürme.

Kod

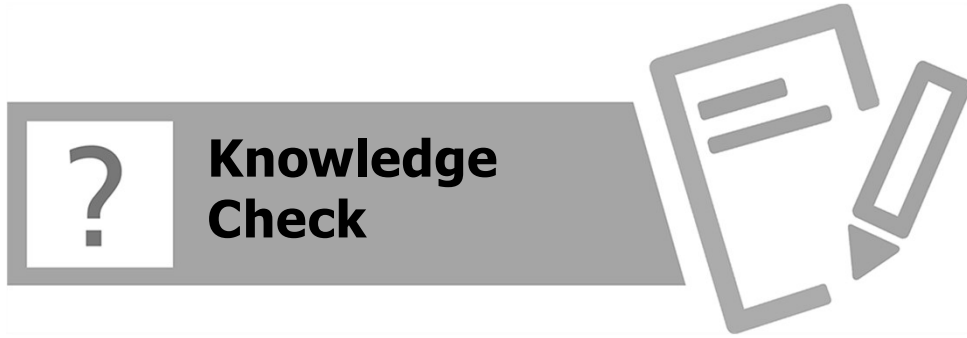
```
aylar = ["ocak", "şubat", "mart", "nisan", "mayıs", "haziran"]  
aylar = tuple(aylar)  
print(aylar)  
aylar=list(aylar)
```

## Demetlerde sıralama

sorted( ) metodu ile öğeleri sıralar.

Kod

```
demet = ("lale", "kardelen", "zambak", "papatya")  
demet=sorted(demet)  
print(demet)
```



1

**5 ve 6 elemanlı olacak şekilde iki adet demet tanımlayınız. Bu iki adet demeti önce birleştirip sonra eleman sayılarını bulan programı yapınız.**



1

**5 ve 6 elemanlı olacak şekilde iki adet demet tanımlayınız. Bu iki adet demeti önce birleştirip sonra eleman sayılarını bulan programı yapınız.**



```
aylar = ("ocak", "şubat", "mart", "nisan", "mayıs", "haziran")
mevsimler = ("kış", "ilkbahar", "yaz", "sonbahar")
yeni_demet = tuple(mevsimler + aylar)
print(len(yeni_demet))
```

Doğru Cevap

2

**Sayılar=(20,24,25,79,40,39,50) demet olarak tanımlanmıştır. Bu demet'te 5'e bölünenleri ekrana yazdıran programı yapınız.**



2

**Sayılar=(20,24,25,79,40,39,50) demet olarak tanımlanmıştır. Bu demet'te 5'e bölünenleri ekrana yazdıran programı yapınız.**



```
sayılar=(20,24,25,79,40,39,50)
for i in sayılar:
    if i%5 ==0:
        print(i)
```

Doğru Cevap

3

**Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?**

```
demet = ("hasan","ali","c","mehmet","deniz","f","fatma")
yenidemet = demet[3:5]
print(yenidemet)
```



3

**Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?**

```
demet = ("hasan","ali","c","mehmet","deniz","f","fatma")  
yenidemet = demet[3:5]  
print(yenidemet)
```



Doğru Cevap

('mehmet', 'deniz')

4

**Uygulama= ("ali","veli","ayşe","Fatma","Hayriye","ali","deniz") şeklinde tanımlanmış bir demet'te ali adlı öğenin kaç adet olduğunu bulunuz.**

4

**Uygulama=("ali","veli","ayşe","Fatma","Hayriye","ali","deniz")** şeklinde tanımlanmış bir demet'te ali adlı öğenin kaç adet olduğunu bulunuz.



```
uygulama=("ali","veli","ayşe","Fatma","Hayriye","ali","deniz")  
print(uygulama.count("ali"))
```

Doğru Cevap

5

**Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?**

```
sozluk = {'renk': 'mavi', 'kıyafet': 'pantolon', 'beden': 'M'}  
for anahtar in sozluk:  
    print(anahtar)
```





5

**Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?**

```
sozluk = {'renk': 'mavi', 'kıyafet': 'pantolon', 'beden': 'M'}  
for anahtar in sozluk:  
    print(anahtar)
```



Doğru Cevap

renk  
kıyafet  
beden

6

**Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?**

```
sozluk = {'renk': 'mavi', 'kıyafet': 'pantolon', 'beden': 'M'}  
for anahtar in sozluk:  
    print(anahtar,sozluk[anahtar])  
    sozluk_bilesenleri=sozluk.items()  
for bilesen in sozluk_bilesenleri :  
    print(bilesen)  
    print(type(bilesen))# demet veri tipinde oluyor
```



6

Aşağıdaki kod çalıştırıldığında ekran çıktısı nedir?

```
sozluk = {'renk': 'mavi', 'kiyafet': 'pantolon', 'beden': 'M'}
for anahtar in sozluk:
    print(anahtar,sozluk[anahtar])
    sozluk_bilesenleri=sozluk.items()
for bilesen in sozluk_bilesenleri :
    print(bilesen)
    print(type(bilesen))# demet veri tipinde oluyor
```



renk mavi  
kiyafet pantolon  
beden M

Doğru Cevap

('renk', 'mavi')  
<class 'tuple'>  
('kiyafet', 'pantolon')  
<class 'tuple'>  
('beden', 'M')  
<class 'tuple'>



# Teşekkürler