



# Bilgisayar Programlama 2

Ders Notu: 7

# Sözlükler (Dictionaries)

- Sözlük, listenin daha genel bir versiyonudur. Örnek, yılın aylarındaki gün sayısını içeren bir liste:

```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

- Ocak ayındaki gün sayısı `days[0]`, Aralık ayı ise `days[11]` veya `days[-1]`'dir.

- Yılın aylarındaki günlerin sözlüğü:

```
days = {'January':31, 'February':28, 'March':31, 'April':30,  
        'May':31, 'June':30, 'July':31, 'August':31,  
        'September':30, 'October':31, 'November':30, 'December':31}
```

- Ocak ayındaki gün sayısını almak için `days['January']` ifadesini kullanıyoruz. Burada sözlük kullanmanın bir avantajı, kodun daha okunabilir olması ve listedeki hangi indekste belirli bir ayın yer aldığını bulmak zorunda kalmamamızdır. Sözlüklerin başka birçok kullanım alanı da vardır.

# Dictionaries

- **Sözlük Oluşturma:** Basit bir sözlük örneği

```
d = {'A':100, 'B':200}
```

- Bir sözlük tanımlamak için onu süslü parantezler içine alırız, {}. Her girdi, iki nokta üst üste ile ayrılmış bir çiftten oluşur. Çiftin ilk kısmı anahtar, ikinci kısmı ise değer olarak adlandırılır. Anahtar, bir indeks gibi davranır.
- Örneğin, ilk çiftte, 'A':100, anahtar 'A', değer 100'dür ve d['A'] 100'ü verir.
- Anahtarlar genellikle string'lerdir, ancak tamsayılar, ondalık sayılar ve daha birçok şey de olabilirler. Aynı sözlükte farklı anahtar türlerini ve farklı değer türlerini de karıştırabilirsiniz.

- **Sözlüklerin değiştirilmesi:**

```
d = {'A':100, 'B':200}
```

- d['A']'yı 400'e değiştirmek için: `d['A']=400`
- Sözlüğe yeni bir madde eklemek için, aşağıdaki gibi atama işlemi yapabiliriz: `d['C']=500`
- Bu tür işlemlerin listelerle çalışmadığını unutmayın. İki elemanlı bir listede L[2]=500 işlemini yapmak, dizin aralığı dışında hatası verecektir. Ancak sözlüklerle çalışır.
- Sözlükten bir öğeyi silmek için `del` operatörünü kullanılır.

```
del d['A']
```

# Dictionary Örnekleri

- **Örnek 1:** Bir sözlüğü, gerçek bir tanımlar sözlüğü olarak kullanabilirsiniz:

```
1 d = {'dog' : 'has a tail and goes woof!'},  
2   'cat' : 'says meow',  
3   'mouse' : 'chased by cats'}
```

- Sözlük kullanımına dair farklı bir örnek:

```
1 d = {'dog' : 'has a tail and goes woof!'},  
2   'cat' : 'says meow',  
3   'mouse' : 'chased by cats'}  
4  
5 word = input('Enter a word: ')  
6 d[word] = 'a new word'  
7  
8 print(d)  
9 print(d['cat'])
```

---

- **Örnek 2:** Aşağıdaki sözlük, Roma rakamlarıyla çalışan bir uygulamada faydalı olacaktır.

```
1 numerals = {'I':1, 'V':5, 'X':10, 'L':50, 'C':100, 'D':500, 'M':1000}
2 print(numerals['X'])
```

- **Örnek 3:** Scrabble oyununda her harfin kendine ait bir puan değeri vardır. Harf değerleri için aşağıdaki sözlüğü kullanabiliriz:

```
1 points = {'A':1, 'B':3, 'C':3, 'D':2, 'E':1, 'F':4, 'G':2,
2 'H':4, 'I':1, 'J':8, 'K':5, 'L':1, 'M':3, 'N':1,
3 'O':1, 'P':3, 'Q':10, 'R':1, 'S':1, 'T':1, 'U':1,
4 'V':4, 'W':4, 'X':8, 'Y':4, 'Z':10}
5
6 word = input("Enter a word: ").upper()
7
8 total = 0
9 for c in word:
10 | total += points[c]
11
12 print(total)
```

- **Sözlük kopyalama:** Liste kopyalamada olduğu gibi, sözlük kopyalama da karmaşık bir işlemdir ve bunun nedenlerini daha sonra ele alınacaktır. Bir sözlüğü kopyalamak için, kopyalama methodu kullanılır:

```
d2 = d.copy()
```

- Operatörde, bir şeyin sözlükte anahtar olup olmadığını belirtmek için kullanılır. Örneğin, aşağıdaki gibi bir sözlüğe sahip olduğu varsayılırsa:

```
d = {'A':100, 'B':200}
```

- Sözlükte bulunmayan bir anahtara başvurmak hataya neden olur. Örneğin, `print(d['C'])` başarısız olur. Bu hatayı önlemek için, bir anahtarı kullanmaya çalışmadan önce, anahtarın sözlükte olup olmadığını kontrol etmek için `'in'` operatörünü kullanabiliriz:

```
1 d = {'A':100, 'B':200}
2
3 harf = input('Bir harf giriniz: ')
4 if harf in d:
5     print('Anahtar değeri: ', d[harf])
6 else:
7     print('Sözlükte yok')
```

- **Sözlüklerde Döngü:** Sözlüklerde döngü oluşturmak, listelerde döngü oluşturmaya benzer. Bir sözlükteki anahtarlarını ve değerlerini sırasıyla yazdıran bir örnek:

```
for key in d:  
    print (key)
```

```
for key in d:  
    print (d[key])
```

- **Anahtar ve değer listeleri:** Aşağıdaki tablo, bir sözlükten anahtar ve değer listeleri elde etmenin yollarını göstermektedir. Bu tabloda `d={'A':1,'B':3}` sözlüğü kullanılmaktadır.

Statement	Result	Description
<code>list(d)</code>	<code>['A', 'B']</code>	keys of d
<code>list(d.values())</code>	<code>[1, 3]</code>	values of d
<code>list(d.items())</code>	<code>[('A', 1), ('B', 3)]</code>	(key,value) pairs of d

- **dict:** 'dict' fonksiyonu, sözlük oluşturmanın başka bir yoludur.

```
d = dict([ ('A', 100), ('B', 300) ])
```

Bu, `{'A':100,'B':300}` sözlüğünü oluşturur.

# Alıştırmalar

## ▪ Soru 1:

- Kullanıcıdan ürün adlarını ve fiyatlarını tekrar tekrar girmesini isteyen bir program yazın. Tüm bu bilgileri, anahtarları ürün adları ve değerleri fiyatları olan bir sözlükte saklayın.
- Kullanıcı ürün ve fiyat girişini tamamladıktan sonra, tekrar tekrar bir ürün adı girmesine izin verin ve ilgili fiyatı veya ürün sözlükte yoksa bir mesaj yazdırın.

```
1  urunler = {}
2
3  while True:
4      urun_adi = input("Ürün adını girin (Çıkmak için 'q' tuşuna basın): ")
5      if urun_adi.lower() == 'q':
6          break
7      fiyat = float(input("Fiyatını girin: "))
8      urunler[urun_adi] = fiyat
9
10 print("\nÜrün girişi tamamlandı.")
11 print("Kaydedilen ürünler:", urunler)
12
13 while True:
14     aranan_urun = input("\nFiyatını öğrenmek istediğiniz ürünü girin (Çıkmak için 'q' tuşuna basın): ")
15     if aranan_urun.lower() == 'q':
16         print("Program sona erdi.")
17         break
18     if aranan_urun in urunler:
19         print(aranan_urun, "ürününün fiyatı:", urunler[aranan_urun])
20     else:
21         print("Bu ürün sözlükte yok.")
```

# Alıştırmalar

- **Soru 2:** Önceki problemde oluşturulan sözlüğü kullanarak, kullanıcının bir dolar tutarı girmesine izin verin ve fiyatı bu tutardan düşük olan tüm ürünleri yazdırın.

```
1  urunler = {'elma': 20.0, 'kiraz': 30.0, 'karpuz': 30.0}
2
3
4  # Kullanıcıdan tutar alma
5  tutar = float(input("\nBir dolar tutari girin: "))
6
7  print("\nBu tutardan daha ucuz ürünler:")
8  for urun in urunler:
9      |   if urunler[urun] < tutar:
10     |       |   print(urun, ":", urunler[urun])
```

# Alıştırmalar

- **Soru 3:** Bu problem için, bu bölümün başındaki, anahtarları ay adları ve değerleri ilgili aylardaki gün sayısı olan sözlüğü kullanın.
- **(a)** Kullanıcıdan bir ay adı girmesini isteyin ve sözlüğü kullanarak o ayda kaç gün olduğunu söyleyin.
- **(b)** Tüm anahtarları alfabetik sırayla yazdırın.
- **(c)** 31 gün olan tüm ayları yazdırın.

```
1  aylar = {
2      "Ocak": 31,
3      "Şubat": 28,
4      "Mart": 31,
5      "Nisan": 30,
6      "Mayis": 31,
7      "Haziran": 30,
8      "Temmuz": 31,
9      "Ağustos": 31,
10     "Eylül": 30,
11     "Ekim": 31,
12     "Kasim": 30,
13     "Aralik": 31
14 }
15
16 # (a) Kullanıcıdan ay adi alip gün sayisini yazdirma
17 ay = input("Bir ay adi girin: ")
18
19 if ay in aylar:
20     print(ay, "ayinda", aylar[ay], "gün vardır.")
21 else:
22     print("Bu ay sözlükte yok.")
23
```

```
23
24 # (b) Tüm anahtarları alfabetik sırayla yazdirma
25 print("\nAylar alfabetik sırayla:")
26 for ay_adi in sorted(aylar):
27     print(ay_adi)
28
29 # (c) 31 gün olan ayları yazdirma
30 print("\n31 gün olan aylar:")
31 for ay_adi in aylar:
32     if aylar[ay_adi] == 31:
33         print(ay_adi)
```

# Alıştırmalar

- **Soru 4:** On adet kullanıcı adı ve şifre içeren bir sözlük kullanan bir program yazın.
- Program, kullanıcıdan kullanıcı adını ve şifresini girmesini istemelidir.
- Kullanıcı adı sözlükte yoksa, program kişinin sistemin geçerli bir kullanıcısı olmadığını belirtmelidir.
- Kullanıcı adı sözlükte varsa ancak kullanıcı doğru şifreyi girmezse, program şifrenin geçersiz olduğunu belirtmelidir. Şifre doğruysa, program kullanıcıya sisteme giriş yaptığını bildirmelidir.

```
1  kullanicilar = {
2      "ali": "1234",
3      "ayse": "abcd",
4      "mehmet": "5678",
5      "zeynep": "pass1",
6      "ahmet": "qwerty",
7      "fatma": "9876",
8      "can": "deneme",
9      "elif": "elif123",
10     "mert": "1111",
11     "selin": "selin22"
12 }
13
14 kullanıcı_adi = input("Kullanıcı adını girin: ")
15 şifre = input("$ifreyi girin: ")
16
17 if kullanıcı_adi not in kullanicilar:
18     print("Bu kişi sistemin geçerli bir kullanıcısı değildir.")
19 else:
20     if şifre == kullanicilar[kullanıcı_adi]:
21         print("Sisteme başarıyla giriş yaptınız.")
22     else:
23         print("Geçersiz şifre.")
..
```

# Alıştırmalar

- **Soru 5:** Kullanıcıdan tekrar tekrar takım adını ve takımın kazandığı ve kaybettiği oyun sayısını girmesini isteyin.
- Bu bilgileri, anahtarları takım adları olan ve değerleri [galibiyetler, mağlubiyetler] biçiminde listeler olan bir sözlükte saklayın.
- Yukarıda oluşturulan sözlüğü kullanarak, kullanıcının bir takım adı girmesine izin verin ve takımın kazanma yüzdesini yazdırın.

```
1  takimler = {}
2
3  # Takim bilgilerini alma
4  while True:
5      takim_adi = input("Takim adini girin (bitirmek için q): ")
6
7      if takim_adi.lower() == "q":
8          break
9
10     galibiyet = int(input("Kazandıđı oyun sayisi: "))
11     maglubiyet = int(input("Kaybettiđı oyun sayisi: "))
12
13     takimler[takim_adi] = [galibiyet, maglubiyet]
14
15 # Takimin kazanma yüzdesini hesaplama
16 aranan_takim = input("\nKazanma yüzdesini görmek istediđiniz takim: ")
17
18 if aranan_takim in takimler:
19     galibiyet = takimler[aranan_takim][0]
20     maglubiyet = takimler[aranan_takim][1]
21     toplam_mac = galibiyet + maglubiyet
22
23     if toplam_mac > 0:
24         yuzde = galibiyet / toplam_mac * 100
25         print(aranan_takim, "takiminin kazanma yüzdesi: %", yuzde)
26     else:
27         print("Bu takim hiç maç yapmamış.")
28 else:
29     print("Bu takim sözlükte yok.")
```