

```
In [1]: # update() metodu sözlükteki tüm anahtarların değerlerini günceller
# tek bir öge üzerinde işlem yapmaz. Tüm sözlük baştan yazılır.
```

```
In [2]: maaslar = {
    'Hüzeyin' : 5000,
    'Barkın' : 4500,
    'Volkan' : 4000
}
maaslar
```

```
Out[2]: {'Hüzeyin': 5000, 'Barkın': 4500, 'Volkan': 4000}
```

```
In [3]: # tüm maaşların hepsine birden değişiklik olsun
yeni_maaslar = {
    'Hüzeyin' : 7000,
    'Barkın' : 7000,
    'Volkan' : 7000
}
yeni_maaslar
```

```
Out[3]: {'Hüzeyin': 7000, 'Barkın': 7000, 'Volkan': 7000}
```

```
In [4]: # maaslar sözlüğündeki verileri yeni_maaslar sözlüğüne göre güncellemek isteyelim
# update() metoduna parametre olarak yeni oluşturulan sözlük verildi
maaslar.update(yeni_maaslar)
maaslar
# böylece maaslar sözlüğündeki tüm maaşlar 7000 olmuştur
# böylelikle yeni sözlükteki bilgiler eski sözlüğü güncellemek için kullanıldı
```

```
Out[4]: {'Hüzeyin': 7000, 'Barkın': 7000, 'Volkan': 7000}
```

```
In [5]: yeni_maaslar
```

```
Out[5]: {'Hüzeyin': 7000, 'Barkın': 7000, 'Volkan': 7000}
```

```
In [6]: sozluk={
    'table' : 'masa',
    'door' : 'kapı',
    'phone' : 'telefon',
    'cable' : 'kablo',
    'mouse' : 'fare'
}
sozluk
```

```
Out[6]: {'table': 'masa',
    'door': 'kapı',
    'phone': 'telefon',
    'cable': 'kablo',
    'mouse': 'fare'}
```

```
In [7]: # items() metodu anahtar-değer ikilisine ulaştırır
# items() metodu tüm anahtar-değer ikililerini verir
sozluk.items()
```

```
Out[7]: dict_items([('table', 'masa'), ('door', 'kapı'), ('phone', 'telefon'), ('cable',
    'kablo'), ('mouse', 'fare')])
```

```
In [8]: type(sozluk.items())
```

```
Out[8]: dict_items
```

```
In [9]: # çıktığı daha anlamlı ve kullanılabilir hale dönüştürelim
# list() metodu ile dict_items veri tipini listeye dönüştürelim
ogeler = list(sozluk.items())
ogeler
```

```
Out[9]: [('table', 'masa'),
         ('door', 'kapı'),
         ('phone', 'telefon'),
         ('cable', 'kablo'),
         ('mouse', 'fare')]
```

```
In [10]: type(ogeler)
```

```
Out[10]: list
```

```
In [11]: # listenin ilk elemanı
ogeler[0]
```

```
Out[11]: ('table', 'masa')
```

```
In [12]: # listeye dönüştürülen sözlüğün her bir elemanı tuple (demet) tipindedir
type(ogeler[0])
# liste içinde 5 adet tuple veri tipi var.
# tuple 0, index anahtar ve 1. index değerdir
# items() metodundan gelen anahtar-değer verileri tuple halindedir
```

```
Out[12]: tuple
```

```
In [13]: # values() metodu ile tüm değerler çekilir
sozluk.values()
```

```
Out[13]: dict_values(['masa', 'kapı', 'telefon', 'kablo', 'fare'])
```

```
In [14]: # keys() metodu ile tüm anahtarlar çekilir
sozluk.keys()
```

```
Out[14]: dict_keys(['table', 'door', 'phone', 'cable', 'mouse'])
```

```
In [15]: type(sozluk.values())
```

```
Out[15]: dict_values
```

```
In [16]: type(sozluk.keys())
```

```
Out[16]: dict_keys
```

```
In [17]: # sözlük değerlerini içeren veriyi degerler isimli listeye dönüştürelim
degerler = list(sozluk.values())
degerler
```

```
Out[17]: ['masa', 'kapı', 'telefon', 'kablo', 'fare']
```

```
In [18]: # sözlük anahtarlarını içeren veriyi anahtarlar isimli listeye dönüştürelim
anahtarlar = list(sozluk.keys())
anahtarlar
```

```
Out[18]: ['table', 'door', 'phone', 'cable', 'mouse']
```

```
In [19]: # fromkeys() metodu yeni sözlük üretmek için kullanılır
# yeni üretilen sözlük için birden fazla anahtar alır
# ancak karşısına koymak için yalnız 1 değer alır
# üretilen sözlükteki anahtarların karşısındaki değerler tamamen aynıdır
meyveler = 'Elma', 'Armut', 'Portakal'
fiyat = '15 TL'

sozluk = dict.fromkeys(meyveler,fiyat)
sozluk
```

```
Out[19]: {'Elma': '15 TL', 'Armut': '15 TL', 'Portakal': '15 TL'}
```

```
In [20]: # bu metod ile anahtarları oluştururken demet ya da listelerden faydalanılır
# meyveler değişkenini tanımlarken parantez kullanmadık, direk virgüllerle ayırdık
# bu da demet oluşturmanın bir başka yöntemidir
```

```
In [21]: # sözlük üreticileri kolayca sözlük üretmemizi sağlar
# sözlük üreticileri kimi durumlarda işlerimizi epey kolaylaştırır
```

```
In [22]: # kullanıcı bir kelime girecek ve bu kelimenin harflerine
# alfabe sırası ile numara verilecek

# alfabemizi bir karakter dizisi olarak tanımlıyoruz
alfabe = "abcçdefgğhıijklmnoöprsstüüvyz"

# kullanıcıdan bir kelime alıyoruz
kelime = input('Kelime:')

# boş bir sözlük oluşturuyoruz
sozluk = {}

# for döngüsü ile bu sözlüğe eleman ekliyoruz
# her döngüde kelime değişkeninde bulunan bir harf sözlüğün anahtarı olur
# harfin alfabe değişkenindeki sırası ise sözlüğün değeri olur
for i in kelime:
    sozluk[i] = alfabe.index(i)

sozluk
```

Kelime:serkan

```
Out[22]: {'s': 21, 'e': 5, 'r': 20, 'k': 13, 'a': 0, 'n': 16}
```

```
In [23]: def sozluk_uret(kelime):
sozluk = {}
alfabe = "abcçdefgğhıijklmnoöprsstüüvyz"

for i in kelime:
    sozluk[i] = alfabe.index(i)

return sozluk
```

```
In [24]: kelime = input('Kelime:')
sozluk_uret(kelime)
```

Kelime:volkan

```
Out[24]: {'v': 26, 'o': 17, 'l': 14, 'k': 13, 'a': 0, 'n': 16}
```

```
In [25]: kelime = input('Kelime:')
sozluk_uret(kelime)
```

Kelime:birecik

```
Out[25]: {'b': 1, 'i': 11, 'r': 20, 'e': 5, 'c': 2, 'k': 13}
```

```
In [26]: # bu işlemi sözlük üreticileri ile yapalım
def sozluk_ureteci(kelime):
    alfabe = "abcçdefgğhıijklmnoöprsştuüvyz"

    sozluk = {i:alfabe.index(i) for i in kelime}

    return sozluk
```

```
In [27]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:serkan

```
Out[27]: {'s': 21, 'e': 5, 'r': 20, 'k': 13, 'a': 0, 'n': 16}
```

```
In [28]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:volkan

```
Out[28]: {'v': 26, 'o': 17, 'l': 14, 'k': 13, 'a': 0, 'n': 16}
```

```
In [29]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:birecik

```
Out[29]: {'b': 1, 'i': 11, 'r': 20, 'e': 5, 'c': 2, 'k': 13}
```

```
In [1]: # bu işlemi sözlük üreticileri ile yapalım
def sozluk_ureteci(kelime):
    alfabe = "abcçdefgğhiijklmnoöprsştuüvyz"

    sozluk = {i:alfabe.index(i) for i in kelime}

    return sozluk
```

```
In [2]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:serkan

```
Out[2]: {'s': 21, 'e': 5, 'r': 20, 'k': 13, 'a': 0, 'n': 16}
```

```
In [3]: def sozluk_uret(kelime):
sozluk = {}
alfabe = "abcçdefgğhiijklmnoöprsştuüvyz"

for i in kelime:
    sozluk[i] = alfabe.index(i)

return sozluk
```

```
In [4]: kelime = input('Kelime:')
sozluk_uret(kelime)
```

Kelime:serkan

```
Out[4]: {'s': 21, 'e': 5, 'r': 20, 'k': 13, 'a': 0, 'n': 16}
```

```
In [5]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:volkan

```
Out[5]: {'v': 26, 'o': 17, 'l': 14, 'k': 13, 'a': 0, 'n': 16}
```

```
In [6]: kelime = input('Kelime:')
sozluk_uret(kelime)
```

Kelime:volkan

```
Out[6]: {'v': 26, 'o': 17, 'l': 14, 'k': 13, 'a': 0, 'n': 16}
```

```
In [7]: kelime = input('Kelime:')
sozluk_ureteci(kelime)
```

Kelime:birecik

```
Out[7]: {'b': 1, 'i': 11, 'r': 20, 'e': 5, 'c': 2, 'k': 13}
```

```
In [8]: kelime = input('Kelime:')
sozluk_uret(kelime)
```

Kelime:birecik

```
Out[8]: {'b': 1, 'i': 11, 'r': 20, 'e': 5, 'c': 2, 'k': 13}
```

```
In [1]: # rehber uygulaması için boş bir sözlük oluşturalım
rehber = {}

# uygulamamız biz 'çık' komutunu vermeden kapanmamalıdır
# bunun için while döngüsü ile sürekli bir döngüde olmak gerekir
# ne zaman 'çık' komutunu vereceğimiz belirsiz olduğundan while kullanılır

while True:
    # print() fonksiyonuyla uygulama her başa döndüğünde işlemleri sıralasın
    print("""
[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık
""")

    secim = input('İşlem Numarası:') # kullanıcı yapacağı işlemi seçer

    if secim == '1': # seçim '1' ise kişi sorgulama
        kisi = input('Sorgulanacak Kişi:') # sorgulanacak kişi girilir

        if kisi in rehber.keys(): # aranan isim rehberde varsa
            print('Aranan {} kişisi rehberde var!'.format(kisi))
            print(rehber.get(kisi))

        else: # aranan isim rehberde yoksa
            print('Aranan {} kişisi rehberde yer almıyor!'.format(kisi))

    elif secim == '2': # seçim '2' ise yeni kişi ekleme
        # anahtar olarak eklenecek kişi girilir
        kisi = input('Eklenecek Kişi:')

        # değer olarak eklenecek numara girilir
        numara = input('Eklenecek Numara:')

        # yeni kişi rehberde ekleme, güncelleme yok
        rehber.setdefault(kisi,numara)

        print('{} kişisi {} numarasıyla rehberde eklendi!'.format(kisi,numara))

    elif secim == '3': # seçim '3' ise kişi silme
        kisi = input('Silinecek Kişi:') # silinecek kişi girilir

        varmi = kisi in rehber.keys() # kullanıcının girdiği isim rehberde var mı?

        if varmi: # ilgili kişi varsa
            print('{} kişisi rehberden silindi!'.format(kisi))
            rehber.pop(kisi) # pop() ile ilgili kişi silinir

        else: # ilgili kişi yoksa
            print('Silinecek {} kişisi rehberde yok!'.format(kisi))

    elif secim == 'q' or secim == 'Q': # seçim 'q' veya 'Q' ise uygulamadan çık
        print('Rehber uygulamasından çıkış yapıldı!')
        break # break ile while döngüsü kırılır döngü başa dönmez

    else: # kullanıcı menüde olmayan bir değer ile seçim yaparsa
        print('Hatalı bir giriş yaptınız! Lütfen tekrar deneyin.')
```

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:2
Eklenecek Kişi:serkan
Eklenecek Numara:1234567890
serkan kişisi 1234567890 numarasıyla rehberde eklendi!

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:1
Sorgulanacak Kişi:serkan
Aranan serkan kişisi rehberde var!
1234567890

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:2
Eklenecek Kişi:volkan
Eklenecek Numara:123456789123
volkan kişisi 123456789123 numarasıyla rehberde eklendi!

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:1
Sorgulanacak Kişi:volkan
Aranan volkan kişisi rehberde var!
123456789123

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:3
Silinecek Kişi:ahmet
Silinecek ahmet kişisi rehberde yok!

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:3
Silinecek Kişi:volkan
volkan kişisi rehberden silindi!

[1] Kişi Sorgula
[2] Kişi Ekle
[3] Kişi Sil
[Q] Çık

İşlem Numarası:1

Sorgulanacak Kişi:volkan
Aranan volkan kişisi rehberde yer almıyor!

- [1] Kişi Sorgula
- [2] Kişi Ekle
- [3] Kişi Sil
- [Q] Çık

İşlem Numarası:q
Rehber uygulamasından çıkış yapıldı!