

8. Tuple içindeki elemanın indeksini bulma: Listelerde olduğu gibi index fonksiyonu kullanılır.

Örnek :

```
birimler = ("bit", "inç", "byte", "hertz", "piksel")
```

```
print(birimler.index("byte"))
```

Ekran Çıktısı: 2

9. Tuple birleştirme: Birden fazla tuple birleştirilerek tek bir tuple'da toplanabilir.

Örnek :

```
birimler = ("bit", "inç", "byte", "hertz", "piksel")
```

```
degerler=(8,256,1024)
```

```
birlestir=birimler+degerler
```

```
print(birlestir)
```

Ekran Çıktısı: ('bit', 'inç', 'byte', 'hertz', 'piksel', 8, 256, 1024)

4.3.5. Dictionary (Sözlük) Veri Tipi

Python programlama dilinde sırasız, değiştirilebilir ve belirli bir konuma sahip koleksiyonlar sözlük olarak adlandırılır. Sözlükler süslü (ya da kırlangıç) parantezler arasına yazılır. Sözlük veri tipinde anahtarlar ve bu anahtarların değerleri vardır. Her anahtardan sonra iki nokta (:) kullanılır ve değer yazılır. Anahtar:değer (key:value) ikilileri virgülle birbirinden ayrılır.

Hatırlatma: Liste veri tipinde **köşeli parantez** [], tuple veri tipinde **normal parantez** (), sözlük veri tipinde ise **süslü parantez** {} kullanılır.

Farklı şekillerde tanımlanabilen sözlük veri tipinin genel kullanımı şu şekildedir:

```
sozluk_adi={anahtar:deger}
```

Örnek :

```
sozluk = {"Mesleğiniz":"Öğrenci", "Alanınız":"Bilişim", "Yaşadığınız Yer":"Ankara" }
```

```
print(sozluk)
```

Ekran Çıktısı: {'Mesleğiniz': 'Öğrenci', 'Alanınız': 'Bilişim', 'Yaşadığınız Yer': 'Ankara'}

Sözlükte sadece anahtarları göstermek için **key** ve **values** fonksiyonları kullanılır.

Örnek :

```
sozluk = {"Mesleğiniz":"Öğrenci", "Alanınız":"Bilişim", "Yaşadığınız Yer":"Ankara" }
```

```
print(sozluk.keys())
```

```
print(sozluk.values())
```

Ekran Çıktısı: dict_keys(['Mesleğiniz', 'Alanınız', 'Yaşadığınız Yer'])

```
dict_values(['Öğrenci', 'Bilişim', 'Ankara'])
```

Sözlük veri tipi ile yapılabilecekler genel olarak şu şekildedir:

1. Sözlük elemanlarına erişim aşağıdaki şekilde yapılmaktadır.

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
print(donanim["Türü"])
```

Ekran Çıktısı: RAM

2. Sözlük içindeki değerleri değiştirebilirsiniz. Aşağıda değer değişimine yönelik bir örnek bulunmaktadır.

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
donanim["Kapasitesi"]="16 GB" #burada 8 GB değeri, 16 GB değeri ile değişti.
print(donanim)
```

Ekran Çıktısı: {'Türü': 'RAM', 'Tipi': 'DDR4', 'Kapasitesi': '16 GB'}

3. Diğer veri tiplerinde olduğu gibi sözlüklerde de bir değer olup olmadığına in operatörü ile bakılabilir.

Örnek :

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
print("Türü" in donanim) #Sözlükte Türü anahtarının olup olmadığı kontrol edilmiştir.
```

Ekran Çıktısı: True

4. Sözlüklerde de uzunluk **len** fonksiyonu ile bulunur. Burada eleman sayısının anahtar-değer ikilileri olarak hesaplanacağını unutmayınız.

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
print(len(donanim))
```

Ekran Çıktısı: 3

5. Sözlüğe daha sonra anahtar-değer ikilileri eklenebilir. Aşağıdaki örnekte ikinci satıra dikkat ediniz.

Örnek :

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
donanim["Hızı"]="2400 MHz" #burada ekleme işlemi yapılmıştır.
print(donanim)
```

Ekran Çıktısı: {'Türü': 'RAM', 'Tipi': 'DDR4', 'Kapasitesi': '8 GB', 'Hızı': '2400 MHz'}

6. Sözlük veri tipinde silme işlemi yapmak için **pop** fonksiyonu kullanılır.

Örnek :

```
donanim = {"Türü":"RAM", "Tipi":"DDR4", "Kapasitesi":"8 GB" }
donanim.pop("Kapasitesi") #burada silme işlemi yapılmıştır.
print(donanim)
```

Ekran Çıktısı: {'Türü': 'RAM', 'Tipi': 'DDR4'}

Sıra Sizde: Aşağıdaki sözlükleri oluşturarak sizlerden istenen işlemleri yapınız.

sozluk = {"Bilim insanı": "Aziz Sancar", "Şair": "Mehmet Akif Ersoy", "Astronom": "Ali Kuşçu" }

- sozluk isimli sözlüğü meslekler isimli başka bir sözlüğe kopyalayınız ve ekrana yazdırınız.
- sozluk isimli sözlüğün değerlerini ekrana yazdırınız.
- sozluk isimli sözlüğü içi boş bir sözlük hâline getiriniz.
- sozluk isimli sözlüğe Matematikçi: Cahit Arf ikilisini ekleyiniz.
- sozluk isimli sözlüğün içinde sanatçı anahtarının olup olmadığını sorgulayınız.
- sozluk isimli sözlüğün bilim insanı anahtarındaki değeri Canan Dağdeviren olarak değiştiriniz.
- sozluk isimli sözlüğün şair anahtarı ile eşleşen değeri ekrana yazdırınız.

onemli_telefonlar = {"Acil Çağrı Merkezi": "112", "Polis İmdat": "155", "Milli Eğitim Bakanlığı İletişim Merkezi": "444 0 632" }

- onemli_bilgiler isimli sözlüğün değerlerini ekrana yazdırınız.
- onemli_bilgiler isimli sözlüğü siliniz.
- onemli_bilgiler isimli sözlükten Acil Çağrı Merkezi anahtarını ve değerini siliniz.
- onemli_bilgiler isimli sözlükte Sağlık Bakanlığı İletişim Merkezi olup olmadığını sorgulayınız.
- onemli_bilgiler isimli sözlüğü içi boş bir sözlük hâline getiriniz.

4.3.6. Set (Küme) Veri Tipi

Python programlama dilinde kullanılan veri tiplerinden biri de **set (küme)** veri tipidir. Sözlükler gibi süslü parantezlerin içine yazılan set veri tipi, sözlüklerden farklı olarak ikili anahtar yapısında değildir. Set veri tipinde elemanlar sırasızdır ve tekrar etmez. Türkçeye küme olarak çevrilen bu veri tipi bir dizi matematiksel işlemin kolaylaştırılmasını sağlar.

Set veri tipinin basit kullanımı şu şekildedir:

```
sayilar = {1, 2, 3, 4, 5} #integer veri tipi tırnak içinde yazılmaz.
```

```
print(sayilar)
```

Ekran Çıktısı: {1, 2, 3, 4, 5}

Set veri tipinde de fonksiyonlar kullanılarak bir dizi işlem yapılabilir. Bu fonksiyonlar genel olarak liste, sözlük ve demet veri tipindeki fonksiyonlarla benzerdir. Aşağıda bu fonksiyonlara bazı örnekler verilmiştir:

- Bir elemanın küme içinde olup olmadığı **in fonksiyonu** ile kontrol edilir.

```
sayilar = {1, 2, 3, 4, 5}
```

```
print(6 in sayilar)
```

Ekran Çıktısı: False

2. Küme veri tipinde eleman eklemek için **add()** fonksiyonu kullanılır.

Örnek :

```
sayilar = {1, 2, 3, 4, 5}
```

```
sayilar.add(6)
```

```
print(sayilar)
```

Ekran Çıktısı: {1, 2, 3, 4, 5, 6}

Tek bir eleman yerine birden fazla eleman eklemek için **update()** fonksiyonu kullanılır.

Örnek :

```
sayilar = {1, 2, 3, 4, 5}
```

```
sayilar.update([6,7,8])
```

```
print(sayilar)
```

Ekran Çıktısı: {1, 2, 3, 4, 5, 6, 7, 8}

3. Set içindeki bir elemanı silmek için **remove()** ya da **discard()** fonksiyonları kullanılır. Her iki fonksiyonun kullanımı aynıdır.

Örnek :

```
sayilar = {1, 2, 3, 4, 5}
```

```
sayilar.discard(3)
```

```
print(sayilar)
```

Ekran Çıktısı: {1, 2, 4, 5}

Önemli Not: Verilen örneklerde sadece integer tipi kullanılmış olsa da set veri tipinde farklı veri tiplerini (aynı kümede integer, string veya float gibi) aynı anda kullanabilirsiniz.

Sıra Sizde: Set veri tipinde kullanılan diğer fonksiyonları araştırınız ve uygulayınız.