

Farklı veri tiplerini (listeler, sözlükler vb.) kullanarak programlar geliştirir.

İç İçe Liste Oluşturma

Python programlama dilinde iç içe liste adı verilen yapı ile bir liste içinde başka listeler de tutulabilir.

```
Örnek : meyveler=["elma", "çilek", "armut"]
alisveris_listesi=["süt", "peynir", meyveler]
print(alisveris_listesi)
```

```
Ekran Çıktısı: ['süt', 'peynir', ['elma', 'çilek', 'armut']]
```

Bu örnekte iki farklı liste bulunmaktadır. Görüldüğü üzere alisveris_listesi içinde meyveler listesi de kullanılmıştır. Ekran çıktısına bakıldığında iki listesinin iç içe kullanıldığı görülmektedir.

5. Elemanın olup olmadığını sorgulama: Tuple veri tipinde de listelerde olduğu gibi in operatörü ile bir elemanın listede olup olmadığı kontrol edilebilir. Eleman tuple'daysa True; yoksa False değerleri üretilir.

Örnek :

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

```
Ekran Çıktısı: True
```

6. Tuple uzunluğunu bulma: len fonksiyonu ile tuple'ın eleman sayısı bulunur.

Örnek :

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

```
Ekran Çıktısı: 5
```

4.3.4. Tuple (Demet) Veri Tipi

Listeler konusunda oluşturulan listeler üzerinden daha sonra değişiklikler yapılabildiğini gördünüz. Tuple veri tipi de listelere oldukça benzemektedir. Aralarındaki temel fark ise tuple veri tipinin tanımlandıktan sonra değişikliğe yani eleman ekleme ya da silmeye izin vermemesidir. Tuple veri tipi ile yapılabilecek işlemler şu şekildedir:

1. Tuple oluşturma: Tuple tanımlaması yapılırken listelerden farklı olarak parantezler kullanılır.

Örnek :

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

```
Ekran Çıktısı: ('bit', 'inç', 'byte', 'hertz', 'piksel')
```

Listelere benzer şekilde tuple oluşturulur ve ekrana yazdırılır.

2. Tuple elemanlarına ulaşma: Listelerdeki gibi indeks kullanılır.

Örnek :

```
birimler = ("bit", "inç", "byte", "hertz", "piksel")
print(birimler[3])
```

```
Ekran Çıktısı: hertz
```

`list, tuple, dict, set, frozenset, range, bytearray, memoryview`

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'}