

Demetler (Tuples)

- Demetler, sıralı bir dizidir ve parantez içinde virgülle ayrılmış elemanlar olarak ifade edilir. (Tuples are an ordered sequence expressed as comma-separated elements within parentheses.)
- Demetler değiştirilemezdir (immutable), yani bir demetin elemanları değiştirilemez. (Tuples are immutable, meaning their elements cannot be changed.)

Listeler (Lists)

- Listeler, köşeli parantezler içinde tanımlanan sıralı dizilerdir. (Lists are ordered sequences represented with square brackets.)
- Listeler değiştirilebilir (mutable) ve elemanları üzerinde değişiklik yapılabilir. (Lists are mutable, allowing modifications to their elements.)

Erişim ve İşlemler

- Hem demetler hem de listeler, indeksleme ile elemanlara erişim sağlar. (Both tuples and lists allow access to elements via indexing.)
- Listelerde eleman eklemek için append ve extend gibi yöntemler kullanılabilir. (Methods like append and extend can be used to add elements to lists.)
- Sözlükler(dictionaries), anahtar-değer çiftlerini (key-value pairs) saklamak için kullanılan bir veri yapısıdır. Sözlüklerde anahtarlar (keys) benzersiz ve değiştirilemez (immutable) olmalıdır. Değerler (values) ise değiştirilebilir (mutable) ve tekrar edilebilir (duplicates) olabilir.
- Bir sözlük oluşturmak için süslü parantezler (curly brackets) kullanırız. Anahtarlar, değerlerin önünde yer alır ve her anahtar, bir değerle (value) iki nokta üst üste (colon) ile ayrılır. Örneğin, bir müzik albümünü temsil eden bir sözlükte, albüm adı anahtar (key) ve çıkış tarihi değer (value) olarak kullanılabilir. Sözlükteki bir değere erişmek için anahtarı kullanarak köşeli parantezler (square brackets) ile yazabiliriz. Örneğin:

```
• album = {  
•     "Back in Black": 1980,  
•     "The Dark Side of the Moon": 1973,  
•     "The Bodyguard": 1992  
• }  
•  
• print(album["Back in Black"]) # 1980
```

- Bu örnekte, "Back in Black" anahtarı ile 1980 değerine erişiyoruz.
- Kümeler(sets), Python'da bir **collection** (koleksiyon) türüdür. Diğer koleksiyon türleri olan **lists** (listeler) ve **tuples** (demetler) gibi, kümeler de farklı Python

türlerini içerebilir. Ancak, kümeler **unordered** (sırasız) oldukları için elemanların pozisyonunu kaydetmezler. Ayrıca, kümelerde yalnızca **unique elements** (benzersiz elemanlar) bulunur; yani bir elemandan yalnızca bir tane olabilir. Kümeleri tanımlamak için **curly brackets** (küme parantezleri) kullanırız. Örneğin, bir küme tanımlamak için {eleman1, eleman2} şeklinde yazabilirim.

- Kümelerle bazı işlemler yapabiliriz. Örneğin, bir kümeye eleman eklemek için **add method** (ekleme metodu) kullanırız. Eğer aynı elemanı iki kez eklemeye çalışırsak, hiçbir şey olmaz çünkü kümelerde tekrar eden elemanlar bulunmaz. Bir elemanı kümeden çıkarmak ise **remove method** (çıkarma metodu) kullanırız. Ayrıca, bir elemanın kümede olup olmadığını kontrol etmek için **in command** (içinde komutu) kullanabiliriz. Eğer eleman kümede varsa, True (doğru) döner; yoksa False (yanlış) döner.