# LİSTS-TUPLES-SETS

* Lists in Python are ordered collections of items where each item is assigned an index, starting from 0.
* Tuples, like lists, are ordered collections of items created with parentheses.
* Tuples, like lists, can contain duplicate elements.
* Tuple unpacking allows for assigning tuple items to variables. The values will be assigned in the order they appear in the tuple.

birthday\_date = (12, "August", 1993)

day, month, year = birthday\_date

print(day) //12

print(month) //"August",

print(year) // 1993

* The \* operator in tuple unpacking is used to gather multiple elements from the tuple into a list. This is useful when dealing with tuples of unknown length.

scores = (98, 96, 91, 88, 64)

winner, \*rest = scores

print(winner) //98

print(rest) //[96, 91, 88, 64]

* Listenin bir değeri başka bir değer ile değiştirilirse o indexteki değer artık yeni değer ile değiştirilir ve index kayması olamaz:

years = [2002, 2008, 1999]

years[1] = 2007

for year in years:

  print(year)

Output:

2002

2007

1999

* İndex kayması olması için insert methodu kullanılmalıdır.

years = [2002, 2008, 1999] years.insert(1, 2007) # 1. indekse 2007 ekle print(years) //[2002, 2007, 2008, 1999]

* Sets, unlike lists and tuples, are unordered collections. They are created with curly brackets { }.
* Sets are unordered and don't support indexing or slicing.
* Sets can't have duplicates, which is very helpful when developers need to ensure that each item in a collection is unique. For example, in social media apps, your friends list should not have duplicates.
* Adding duplicate items to a set doesn't cause an error; instead, it's ignored.

friends = {'Anna', 'Mery', 'Mery', 'Jonathan'}

print(friends) //{'Anna', 'Mery', 'Jonathan'}

* Like lists and tuples, sets can have values with different data types.
* Sets are mutable, meaning you can add or remove items from them.
* Use the **add()** and **remove()**functions, each with a value as an argument, to add or remove it from a set.

guests = {'Anna', 'Mery', 'Jonathan'}

#adding 'Robert'

guests.add('Robert')

#removing 'Mery'

guests.remove('Mery')

print(guests) //{'Anna', 'Jonathan', 'Robert'}

* The **append()** function works only with ordered collection types, like lists, and adds an item to the end of the collection. Sets are unordered, that's why you can't use it on them.
* The **clear()** function doesn't accept an argument and removes all the items from a set.
* The **union()** function called returns a new set with all elements from both sets, omitting duplicates.

set1 = {'apple', 'banana'}

set2 = {'banana', 'cherry'}

combined\_set = set1.union(set2)

print(combined\_set) //{'cherry', 'apple', 'banana'}

* The **difference()** function returns a set containing elements that are only in the first set and not in the second.

set1 = {'apple', 'banana', 'cherry'}

set2 = {'banana', 'orange'}

unique = set1.difference(set2)

print(unique) // {'cherry', 'apple'}

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu