

March 3 - DATA STRUCTURES

Data structure - user will define the values more than one. We can say collection of data types - List - Tuple - Set - Dict

LISTS

- Lists are used to store multiple items in a single variable.
- it allows duplicate values, Since lists are indexed, lists can have items with the same value:
- it allows different data types. List items can be of any data type.
- List items are ordered, changeable, and allow duplicate values.
- List items are indexed, the first item has index [0], the second item has index [1] etc.
- When we say that lists are ordered, it means that the items have a defined order, and that order will not change.
- But there are some list methods that will change the order, but in general: the order of the items will not change.

```
In [7]: l = []  
print(l)  
type(l) # it will print type of the list
```

```
[]
```

```
Out[7]: list
```

```
In [8]: len(l) # To determine how many items a list has, use the len() function:
```

```
Out[8]: 0
```

List methods are as below

- append - done (appends an element to list at the end)
- copy - done (copy a list from one to another list)
- count - done (count how many times value is in the list)
- remove - done (remove a specified value in the list)
- clear - done (removed all the values in the list)
- extend
- index
- insert
- pop
- reverse
- sort

```
In [10]: l.append(10)
```

```
In [11]: 1
```

```
Out[11]: [10, 10]
```

```
In [12]: len(1)
```

```
Out[12]: 2
```

```
In [14]: 1.append(20)
1.append(30)
1.append(40)
1.append(50)
```

```
In [15]: len(1)
```

```
Out[15]: 6
```

```
In [16]: 1
```

```
Out[16]: [10, 10, 20, 30, 40, 50]
```

```
In [17]: # List items are indexed and you can access them by referring to the index number
1[0]
1[1]
```

```
Out[17]: 10
```

```
In [22]: ''' Negative indexing means start from the end

-1 refers to the last item, -2 refers to the second last item etc.'''
print(1[-1])
1[-2]
```

```
50
```

```
Out[22]: 40
```

```
In [15]: id(1) # Location in memory
```

```
Out[15]: 2297448036288
```

List Slicing

```
In [23]: 1[:] # it will give all items in the list usually syntax for this is list_name[s
```

```
Out[23]: [10, 10, 20, 30, 40, 50]
```

```
In [46]: 1[1]
```

```
Out[46]: 10
```

```
In [25]: 1[-1]
```

```
Out[25]: 50
```

```
In [26]: l[-3]
```

```
Out[26]: 30
```

```
In [27]: l1 = l.copy()
```

```
In [28]: l1
```

```
Out[28]: [10, 10, 20, 30, 40, 50]
```

```
In [29]: l == l1
```

```
Out[29]: True
```

```
In [32]: l1.append(2.3)
l1.append(True)
l1.append(1+7j)
```

```
In [26]: l1.append(1+7j)
```

```
In [33]: l1
```

```
Out[33]: [10, 10, 20, 30, 40, 50, 2.3, True, (1+7j)]
```

```
In [34]: l1.append(2.3)
l1.append(True)
```

```
In [35]: l1
```

```
Out[35]: [10, 10, 20, 30, 40, 50, 2.3, True, (1+7j), 2.3, True]
```

```
In [36]: l1.count(20)
```

```
Out[36]: 1
```

```
In [37]: l
```

```
Out[37]: [10, 10, 20, 30, 40, 50]
```

```
In [38]: l1
```

```
Out[38]: [10, 10, 20, 30, 40, 50, 2.3, True, (1+7j), 2.3, True]
```

```
In [39]: l2
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[39], line 1
----> 1 l2

NameError: name 'l2' is not defined
```

```
In [40]: l2 = l1.copy()
```

```
In [41]: l2
```

Out[41]: [10, 10, 20, 30, 40, 50, 2.3, True, (1+7j), 2.3, True]

In [42]: `l2.remove(20)`

In [43]: `l2`

Out[43]: [10, 10, 30, 40, 50, 2.3, True, (1+7j), 2.3, True]

In [44]: `l2.clear()` *# it removes all the items in the list.*

In [45]: `l2`

Out[45]: []

In []: