

# Data Structures

## Basic Data Structures in python

### 1-Tuple

### 2-List

### 3-Dictionary

### 4-Set

## 1-Tuple

- ordered collection of element.
- enclosed in () in round braces/paranthesis.
- different kind of elements can be stored.
- once element are stored you can not change them (immutable).
- mutable...something that can be changed,-unmutable.....something that cannot be changed.

```
In [1]: tup1=(1,"java",True,2.5)
        tup1
```

```
Out[1]: (1, 'java', True, 2.5)
```

```
In [2]: type(tup1)
```

```
Out[2]: tuple
```

```
In [3]: tup1[1]
```

```
Out[3]: 'java'
```

```
In [4]: tup1[0:4] #last element is exclusive
```

```
Out[4]: (1, 'java', True, 2.5)
```

```
In [5]: # count of element in tuples
        len(tup1)
```

```
Out[5]: 4
```

```
In [6]: tup2=(117,"Muhammad Shahzeb",2.8,False)
        tup2
```

Out[6]: (117, 'Muhammad Shahzeb', 2.8, False)

```
In [7]: # concatenation in tuple  
        tup1+tup2    #concatenate the elements of both tuples
```

Out[7]: (1, 'java', True, 2.5, 117, 'Muhammad Shahzeb', 2.8, False)

```
In [8]: tup1*2
```

Out[8]: (1, 'java', True, 2.5, 1, 'java', True, 2.5)

```
In [9]: tup3=(89,111,116,117)  
        tup3
```

Out[9]: (89, 111, 116, 117)

```
In [10]: max(tup3)
```

Out[10]: 117

```
In [11]: min(tup3)
```

Out[11]: 89

```
In [12]: tup3*2  
        # adding or multiplication in tuple does not add or multiply te elements how ever it jus
```

Out[12]: (89, 111, 116, 117, 89, 111, 116, 117)

---

## 2-List

- ordered collection of element.
- enclosed in [] in square braces.
- different kind of elemenets can be stored.
- once element are stored you can change them as well (mutable).

```
In [13]: list1=[1,"java",True,2.5]  
        list1
```

Out[13]: [1, 'java', True, 2.5]

```
In [14]: type(list1)
```

Out[14]: list

```
In [15]: len(list1)
```

```
Out[15]: 4
```

```
In [16]: list1[2]
```

```
Out[16]: True
```

```
In [17]: list2=[117,"Muhammad Shahzeb",2.8,False]
list2
```

```
Out[17]: [117, 'Muhammad Shahzeb', 2.8, False]
```

```
In [18]: # Concatenation in Lists
list1+list2
```

```
Out[18]: [1, 'java', True, 2.5, 117, 'Muhammad Shahzeb', 2.8, False]
```

```
In [19]: list2*3
```

```
Out[19]: [117,
'Muhammad Shahzeb',
2.8,
False,
117,
'Muhammad Shahzeb',
2.8,
False,
117,
'Muhammad Shahzeb',
2.8,
False]
```

```
In [20]: # we can use different built in function to manipulate the data in list
# name.(then press tab ,the available function will pop up)
# reversing the list. it reverse the elements in list and if we run twice it will also r
list2.reverse()
list2
```

```
Out[20]: [False, 2.8, 'Muhammad Shahzeb', 117]
```

```
In [21]: list1.append("Ali")
list1
```

```
Out[21]: [1, 'java', True, 2.5, 'Ali']
```

```
In [22]: list3=[23,345,45,23,12,234,45,4,23,45,2,34]
len(list3)
```

Out[22]: 12

```
In [23]: list3.sort()  
list3
```

Out[23]: [2, 4, 12, 23, 23, 23, 34, 45, 45, 45, 234, 345]

```
In [24]: list1+list3
```

Out[24]: [1, 'java', True, 2.5, 'Ali', 2, 4, 12, 23, 23, 23, 34, 45, 45, 45, 234, 345]

```
In [26]: list4=list1+list3
```

---

## 3-Dictionary

- an unordered collection of element.
- enclosed in {} in curley braces.
- data stored in Key and Value.
- different kind of elements can be stored.
- once elements are stored you can change them as well (mutable).

```
In [30]: # Food and their prices  
food1={"Samosa":30,"Pakora":100,"Raita":20,"Salad":50,"Chicken Rolls":30}  
food1
```

Out[30]: {'Samosa': 30, 'Pakora': 100, 'Raita': 20, 'Salad': 50, 'Chicken Rolls': 30}

```
In [31]: type(food1)
```

Out[31]: dict

```
In [32]: # extracting data from dictionary  
food1.keys()
```

Out[32]: dict\_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Rolls'])

```
In [34]: food1.values()
```

Out[34]: dict\_values([30, 100, 20, 50, 30])

```
In [36]: # adding a new element in dictionary  
  
food1["Chicken Samosa"]=40  
food1
```

```
Out[36]: {'Samosa': 30,  
         'Pakora': 100,  
         'Raita': 20,  
         'Salad': 50,  
         'Chicken Rolls': 30,  
         'Chicken Samosa': 40}
```

```
In [37]: # update the values  
food1["Chicken Samosa"]=35  
food1
```

```
Out[37]: {'Samosa': 30,  
         'Pakora': 100,  
         'Raita': 20,  
         'Salad': 50,  
         'Chicken Rolls': 30,  
         'Chicken Samosa': 35}
```

```
In [39]: food2={"Date":50,"Chocolates":200,"Sawwayan":300}  
food2
```

```
Out[39]: {'Date': 50, 'Chocolates': 200, 'Sawwayan': 300}
```

```
In [41]: # concatenate  
# food1+food2  
# above written method is not supported in dictionaries so
```

```
In [43]: # to concatenate or update value we will use this method because its unordered element  
food1.update(food2)  
food1
```

```
Out[43]: {'Samosa': 30,  
         'Pakora': 100,  
         'Raita': 20,  
         'Salad': 50,  
         'Chicken Rolls': 30,  
         'Chicken Samosa': 35,  
         'Date': 50,  
         'Chocolates': 200,  
         'Sawwayan': 300}
```

---

## 4-Sets

- an unordered and unindexed collection of element.
- enclosed in {} in curly braces.
- no duplicates are allowed.
- different kind of elements can be stored.
- once elements are stored you can change them as well (mutable).

```
In [46]: s1={1,2,4,5,"Jhelum","Pakistan"}  
s1
```

Out[46]: {1, 2, 4, 5, 'Jhelum', 'Pakistan'}

```
In [47]: # to add any vlaue in sets  
s1.add("Hamid")  
s1
```

Out[47]: {1, 2, 4, 5, 'Hamid', 'Jhelum', 'Pakistan'}

```
In [49]: # Booleans are not allowed to enter in sets  
s1.add(True)  
s1
```

Out[49]: {1, 2, 4, 5, 'Hamid', 'Jhelum', 'Pakistan'}

```
In [52]: # we can not add similar elements twice in sets / duplllication is not allwwed  
s1.add("Hamid")  
s1
```

Out[52]: {1, 2, 4, 5, 'Hamid', 'Jhelum', 'Pakistan'}

```
In [53]: # to remove something from set  
s1.remove(2)  
s1
```

Out[53]: {1, 4, 5, 'Hamid', 'Jhelum', 'Pakistan'}

```
In [55]: # suggestion: for better understanding try to explore all the methods in tuple, list and
```

---