

- Indexing

```
In [1]: ## Make a string  
a = "Samosa Pakora"  
a
```

```
Out[1]: 'Samosa Pakora'
```

```
In [2]: a
```

```
Out[2]: 'Samosa Pakora'
```

```
In [3]: a[0]
```

```
Out[3]: 'S'
```

Each alphabet in the above string "Samosa Pakora" is an index you can call each index by its number

```
In [5]: a[3]
```

```
Out[5]: 'o'
```

```
In [ ]: ## **length of indexes**
```

```
In [6]: len(a)
```

```
Out[6]: 13
```

```
In [7]: a[0:5]  
# Last element is exclusive  
# Agar 12 indexes ki string hai to, saari string print krnay k liye [0:13] likhna hoga
```

```
Out[7]: 'Samos'
```

```
In [8]: a[0:6]
```

```
Out[8]: 'Samosa'
```

```
In [9]: a[-2]
```

```
Out[9]: 'r'
```

```
In [10]: a[-6:13]
```

```
Out[10]: 'Pakora'
```

```
In [11]: a[-6:-1]
```

```
Out[11]: 'Pakor'
```

```
In [18]: food = "biryani"  
         food
```

```
Out[18]: 'biryani'
```

String Methods

```
In [19]: food
```

```
Out[19]: 'biryani'
```

```
In [16]: len(food)
```

```
Out[16]: 7
```

```
In [20]: food.capitalize()
```

```
Out[20]: 'Biryani'
```

```
In [21]: food.upper()
```

```
Out[21]: 'BIRYANI'
```

```
In [22]: food.lower()
```

```
Out[22]: 'biryani'
```

```
In [23]: food.replace("b", "Sh")
```

```
Out[23]: 'Shiryani'
```

```
In [26]: # Counting a sepcific alohabet in a string  
         name = "Baba aammar with Dr. Aammar Tufail"  
         name
```

```
Out[26]: 'Baba aammar with Dr. Aammar Tufail'
```

```
In [27]: name.count("a")
```

```
Out[27]: 8
```

- Finding an index numner in string

```
In [28]: name = "Baba aammar with Dr. Aammar Tufail"  
         name.find("T")
```

```
Out[28]: 28
```

- How to split a string

```
In [29]: food = "I love samosa, pakora, biryani, raita and karahi"  
         food
```

```
Out[29]: 'I love samosa, pakora, biryani, raita and karahi'
```

```
In [30]: food.split(",")
```

```
Out[30]: ['I love samosa', ' pakora', ' biryani', ' raita and karahi']
```

Baisc data structures in Python

1- Tuples

2- List

3- Dictionaries

4- Set

- 1 Tuples

- Ordered collection of elements
- Enclosed in ()
- Different kind of elements can be stored
- Once elements are stored you cannot change it (Unmutateable)

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

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

```
In [2]: # type of a tuple  
type(tup1)
```

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

- Indexing in tuple

```
In [4]: tup1[1]
```

```
Out[4]: 'python'
```

```
In [5]: #Last element is exclusive  
tup1[0:3]
```

```
Out[5]: (1, 'python', True)
```

```
In [7]: #count of elements in tuple  
len(tup1)
```

```
Out[7]: 4
```

```
In [8]: tup2 = (2, "babaaammar", 3.5, False)
tup2
```

```
Out[8]: (2, 'babaaammar', 3.5, False)
```

```
In [9]: #concatinate
tup1 + tup2
```

```
Out[9]: (1, 'python', True, 2.5, 2, 'babaaammar', 3.5, False)
```

```
In [10]: tup1 * 2 + tup2
```

```
Out[10]: (1, 'python', True, 2.5, 1, 'python', True, 2.5, 2, 'babaaammar', 3.5, False)
```

```
In [11]: tup3 = (10,20,30,40,50)
tup3
```

```
Out[11]: (10, 20, 30, 40, 50)
```

```
In [14]: #minimum
min(tup3)
```

```
Out[14]: 10
```

```
In [15]: #maximum
max(tup3)
```

```
Out[15]: 50
```

```
In [16]: tup3*2
```

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

- List

- Ordered collection of elements
- Enclosed in [] Square braces
- Mutatable (You can change the Values)

```
In [18]: list1 = [1, "python", True, 3]
list1
```

```
Out[18]: [1, 'python', True, 3]
```

```
In [19]: type(list1)
```

```
Out[19]: list
```

```
In [20]: len(list1)
```

```
Out[20]: 4
```

```
In [21]: list1[2]
```

```
Out[21]: True
```

```
In [22]: list2 = [3, 5, "Aammar", "Codanics", 478, 53.2, False]
list2
```

```
Out[22]: [3, 5, 'Aammar', 'Codanics', 478, 53.2, False]
```

```
In [23]: list1 + list2
```

```
Out[23]: [1, 'python', True, 3, 3, 5, 'Aammar', 'Codanics', 478, 53.2, False]
```

```
In [25]: list1 * 2
```

```
Out[25]: [1, 'python', True, 3, 1, 'python', True, 3]
```

```
In [26]: list1.reverse()
list1
```

```
Out[26]: [3, True, 'python', 1]
```

```
In [27]: #To add in list we use this function Append
list1.append("Codanics Youtube Channel")
list1
```

```
Out[27]: [3, True, 'python', 1, 'Codanics Youtube Channel']
```

```
In [28]: list3 = [10,40,3343,2023,2032,12,8878,2822]
list3
```

```
Out[28]: [10, 40, 3343, 2023, 2032, 12, 8878, 2822]
```

```
In [29]: #sorting a list
list3.sort()
list3
```

```
Out[29]: [10, 12, 40, 2023, 2032, 2822, 3343, 8878]
```

```
In [30]: list3*3
```

```
Out[30]: [10,
12,
40,
2023,
2032,
2822,
3343,
8878,
10,
12,
40,
2023,
2032,
2822,
3343,
8878,
```

```
10,  
12,  
40,  
2023,  
2032,  
2822,  
3343,  
8878]
```

```
In [31]: #using count function in List  
fruits = ["Apple", "Mango", "Banana", "Cherry" , "Papaya"]  
# printing count using count() function  
fruits.count("Apple")
```

```
Out[31]: 1
```

```
In [33]: lists = list1 + list2  
lists
```

```
Out[33]: [3,  
True,  
'python',  
1,  
'Codanics Youtube Channel',  
3,  
5,  
'Aammar',  
'Codanics',  
478,  
53.2,  
False]
```

- Dictionaries

- An unirdered collection of elements
- Key & Value
- Curly braces or brackets {}
- Mutateable

```
In [36]: #foor and their prices  
food1 = {"Samosa": 30, "Pakora": 100, "Raita":20, "Salad":50, "Chicken Rolls":30}  
#samosa is a key and 30 is its value  
food1
```

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

```
In [37]: type(food1)
```

```
Out[37]: dict
```

```
In [40]: #Extract data  
keys1 = food1.keys()  
keys1
```

```
Out[40]: dict_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Rolls'])
```

```
In [41]: values1 = food1.values()
         values1
```

```
Out[41]: dict_values([30, 100, 20, 50, 30])
```

```
In [43]: #adding new element
         food1["Tikki"]=10
         food1
```

```
Out[43]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 10}
```

```
In [44]: #update the values
         food1["Tikki"]=15
         food1
```

```
Out[44]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 15}
```

```
In [45]: food2 = {"Dates":50, "Choclates":200, "Swayyan":100}
         food2
```

```
Out[45]: {'Dates': 50, 'Choclates': 200, 'Swayyan': 100}
```

```
In [48]: #concatenate
         #food1 + food2 (ye iss trha ni hoga, dictionary ko concate krny ka tareeqa kuch aesy ha
         food1.update(food2)
         food1
```

```
Out[48]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 15,
          'Dates': 50,
          'Choclates': 200,
          'Swayyan': 100}
```

- Sets

- Unordered and Unindexed
- Curly braces are used {}
- No duplicates allowed

```
In [50]: s1 = {1, 2,3,5.3, "Aammar", "Codanics", "Faisalabad", True}
         s1
         #You cannot add Boolean Operator it could not be add
```

Out[50]: {1, 2, 3, 5.3, 'Aammar', 'Codanics', 'Faisalabad'}

```
In [53]: s1.add("Aammar")  
s1  
#If you add duplicate it could not be printed twice
```

Out[53]: {1, 2, 3, 5.3, 'Aammar', 'Codanics', 'Faisalabad'}

```
In [54]: s1.add("Aammar1")  
s1
```

Out[54]: {1, 2, 3, 5.3, 'Aammar', 'Aammar1', 'Codanics', 'Faisalabad'}

```
In [56]: s1.remove("Aammar1")  
s1
```

Out[56]: {1, 2, 3, 5.3, 'Codanics', 'Faisalabad'}