

- Indexing

```
In [2]: a = "Samosa Pakora"  
a
```

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

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

```
Out[3]: 'Samosa Pakora'
```

```
In [4]: a[0]
```

```
Out[4]: 'S'
```

```
In [5]: a[1]
```

```
Out[5]: 'a'
```

Length of Indices

```
In [7]: len(a)
```

```
Out[7]: 13
```

```
In [8]: a[12]
```

```
Out[8]: 'a'
```

```
In [9]: a[0:5]
```

```
Out[9]: 'Samos'
```

```
In [10]: # Last index is exclusive  
a[0:6]
```

```
Out[10]: 'Samosa'
```

```
In [11]: a[0:13]
```

```
Out[11]: 'Samosa Pakora'
```

```
In [12]: a[-2]
```

```
Out[12]: 'r'
```

```
In [13]: a[-6:-1]
```

```
Out[13]: 'Pakor'
```

```
In [14]: a[-6:13]
```

```
Out[14]: 'Pakora'
```

```
In [15]: a[-6:len(a)]
```

```
Out[15]: 'Pakora'
```

string methods

```
In [17]: food = "biryani"  
food
```

```
Out[17]: 'biryani'
```

```
In [18]: len(food)
```

```
Out[18]: 7
```

```
In [19]: food.capitalize()
```

```
Out[19]: 'Biryani'
```

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

```
Out[20]: 'BIRYANI'
```

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

```
Out[21]: 'biryani'
```

```
In [22]: food.replace("b", "Sh")
```

```
Out[22]: 'Shiryani'
```

Counting a specific alphabet in a string

```
In [24]: name = "my name is Mumtaz Amir"  
name
```

```
Out[24]: 'my name is Mumtaz Amir'
```

```
In [25]: name.count("i")
```

```
Out[25]: 2
```

- Finding an index number in a string

```
In [27]: text = "I am a good boy"  
text
```

```
Out[27]: 'I am a good boy'
```

```
In [28]: text.find("b")
```

```
Out[28]: 12
```

- How to split a string

```
In [30]: food = "I love samosa, pakora, biryani, raita, karahi"  
food
```

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

```
In [31]: food.split(",")
```

```
Out[31]: ['I love samosa', ' pakora', ' biryani', ' raita', ' karahi']
```

Basic data structures in Python

1. Tuple
2. List
3. Dictionaries
4. Set

1- Tuple

- Ordered collection of elements
- Enclosed in () Paranthesis
- Different kinds of elements can be stored
- Once stored elements cant be changed (immutatble)

```
In [34]: tup1 = (5, "asad", 7.5, False)  
tup1
```

```
Out[34]: (5, 'asad', 7.5, False)
```

```
In [35]: type(tup1)
```

```
Out[35]: tuple
```

```
In [36]: tup1[2]
```

```
Out[36]: 7.5
```

```
In [37]: tup1[0:2]
```

```
Out[37]: (5, 'asad')
```

```
In [38]: len(tup1)
```

```
Out[38]: 4
```

```
In [39]: tup2 = (2, "baba", True, 3.5, "Amir")
tup2
```

```
Out[39]: (2, 'baba', True, 3.5, 'Amir')
```

```
In [40]: # Concatenate tuples
tup1 + tup2
```

```
Out[40]: (5, 'asad', 7.5, False, 2, 'baba', True, 3.5, 'Amir')
```

```
In [41]: # Repeat and Concatenate tuples
tup1*2 + tup2
```

```
Out[41]: (5, 'asad', 7.5, False, 5, 'asad', 7.5, False, 2, 'baba', True, 3.5, 'Amir')
```

```
In [42]: tup3 = (20, 50, 30, 89, 10)
tup3
```

```
Out[42]: (20, 50, 30, 89, 10)
```

```
In [43]: # min value
min(tup3)
```

```
Out[43]: 10
```

```
In [44]: # max value
max(tup3)
```

```
Out[44]: 89
```

```
In [45]: tup3*2
```

```
Out[45]: (20, 50, 30, 89, 10, 20, 50, 30, 89, 10)
```

2- Lists

- Ordered collection of elements

- Enclosed in [] square brackets
- You can change elements (mutable)

```
In [48]: list1 = [5, "asad", 7.5, False]
list1
```

```
Out[48]: [5, 'asad', 7.5, False]
```

```
In [49]: type(list1)
```

```
Out[49]: list
```

```
In [50]: len(list1)
```

```
Out[50]: 4
```

```
In [51]: list1*3
```

```
Out[51]: [5, 'asad', 7.5, False, 5, 'asad', 7.5, False, 5, 'asad', 7.5, False]
```

```
In [52]: list2 = ["Hello", 5, 7.9, "Amir", True]
list2
```

```
Out[52]: ['Hello', 5, 7.9, 'Amir', True]
```

```
In [53]: list1 + list2
```

```
Out[53]: [5, 'asad', 7.5, False, 'Hello', 5, 7.9, 'Amir', True]
```

```
In [54]: list1.reverse()
list1
```

```
Out[54]: [False, 7.5, 'asad', 5]
```

```
In [55]: list1.append("SherDil")
list1
```

```
Out[55]: [False, 7.5, 'asad', 5, 'SherDil']
```

```
In [56]: list1.count('Sherdil')
```

```
Out[56]: 0
```

3- Dictionaries

- An unordered collection of elements
- key and value
- {} curly braces
- mutable/changeable elements

```
In [58]: # Food and its prices
food1 = {"Samosa" : 30, "Pakora" : 100, "Raita": 20, "Salad" : 50, "Chicken Rolls"
food1
```

```
Out[58]: {'Samosa': 30, 'Pakora': 100, 'Raita': 20, 'Salad': 50, 'Chicken Rolls': 30}
```

```
In [59]: type(food1)
```

```
Out[59]: dict
```

```
In [60]: keys1 = food1.keys()
keys1
```

```
Out[60]: dict_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Rolls'])
```

```
In [61]: values1 = food1.values()
values1
```

```
Out[61]: dict_values([30, 100, 20, 50, 30])
```

```
In [62]: # Adding new element
food1["Tikki"] = 10
food1
```

```
Out[62]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 10}
```

```
In [63]: # Updating values
food1["Tikki"] = 15
food1
```

```
Out[63]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 15}
```

```
In [64]: food2 = {"Dates":50, "Chocolates":500, "Fruit": 800}
food2
```

```
Out[64]: {'Dates': 50, 'Chocolates': 500, 'Fruit': 800}
```

```
In [65]: # Concatenate Dict
food1.update(food2)
food1
```

```
Out[65]: {'Samosa': 30,
          'Pakora': 100,
          'Raita': 20,
          'Salad': 50,
          'Chicken Rolls': 30,
          'Tikki': 15,
          'Dates': 50,
          'Chocolates': 500,
          'Fruit': 800}
```

4- Sets

- Unordered and unindexed
- Used {} curly brackets
- Unique values/ No duplicates
- Boolean operators can not be added to sets

```
In [67]: s1 = {1,2,3,"Faisalabad",5.2, "Bhakkar", True}
s1
```

```
Out[67]: {1, 2, 3, 5.2, 'Bhakkar', 'Faisalabad'}
```

```
In [68]: s1.add("Mumtaz")
s1
```

```
Out[68]: {1, 2, 3, 5.2, 'Bhakkar', 'Faisalabad', 'Mumtaz'}
```

```
In [69]: s1.remove("Faisalabad")
s1
```

```
Out[69]: {1, 2, 3, 5.2, 'Bhakkar', 'Mumtaz'}
```

```
In [70]: s1.pop()
s1
```

```
Out[70]: {2, 3, 5.2, 'Bhakkar', 'Mumtaz'}
```

```
In [71]: s2 = {7, "Neelam", "Mumtaz", 3, 8}
s2
```

```
Out[71]: {3, 7, 8, 'Mumtaz', 'Neelam'}
```

```
In [72]: s1.difference(s2)
```

```
Out[72]: {2, 5.2, 'Bhakkar'}
```

```
In [73]: s2.difference(s1)
```

```
Out[73]: {7, 8, 'Neelam'}
```

```
In [74]: s1.intersection(s2)
```

Out[74]: {3, 'Mumtaz'}

In [75]: `s2.intersection(s1)`

Out[75]: {3, 'Mumtaz'}