- Indexing

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

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

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

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

In [3]:
```
# Length of indeces
len(a)
```

Out[3]:
```
13
```

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

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

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

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

In [6]:
```
a[3]
```

Out[6]:
```
'o'
```

In [7]:
```
a[12]
```

Out[7]:
```
'a'
```

In [8]:
```
a[0:5]
```

Out[8]:
```
'Samos'
```

In [9]:
```
# last index s exclusive
a[0:13]
```

Out[9]:
```
'Samosa Pakora'
```

In [10]:
```
a[-2]
```

Out[10]:
```
'r'
```

In [11]:
```
a[-6:len(a)]
```

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

In [12]:
```
food = "biryani"
food
```

Out[12]:
```
'biryani'
```

Strings Methods

In [13]:
```
food
```

Out[13]:
```
'biryani'
```

In [14]:
```
len(food)
```

Out[14]:
```
7
```

In [15]:
```
#capitalize every element
food.capitalize()
```

Out[15]:
```
'Biryani'
```

In [16]:
```
#uppercase letters
food.upper()
```

Out[16]:
```
'BIRYANI'
```

In [17]:
```
#lowercase Letters
food.lower()
```

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

In [18]:
```
food.replace("b", "Sh")
```

Out[18]:
```
'Shiryani'
```

In [19]:
```
# counting a specific alphabet in a string
name = "baba _ammar with Dr Ammar Tufail"
name
```

Out[19]:
```
'baba _ammar with Dr Ammar Tufail'
```

In [20]:
```
name.count("a")
```

Out[20]:
```
6
```

- finding an index number in string

In [21]:
```
name = "baba _ammar with Dr Ammar Tufail"
name
```

Out[21]:
```
'baba _ammar with Dr Ammar Tufail'
```

In [22]:
```
name.find("D")
```

Out[22]:
```
17
```

In [23]:
```
### - how to split a string
food = "I love samosa, pakora, raita, biryan and karahi"
food
```

Out[23]:
```
'I love samosa, pakora, raita, biryan and karahi'
```

In [24]:
```
food.split(",")
```

Out[24]:
```
['I love samosa', ' pakora', ' raita', ' biryan and karahi']
```

Basic Data Structure in Python

1- Tuple

2- List

3- Dictionaries

4- Set

1- Tuple

- ordered collection of elements
- enclosed in () round braces / parentheses
- Different kind of elements can be stored
- Once elements are stored you can not change them (unmutatable)

In [25]:
```
tup1 = (1, "Python", True, 2.5)
tup1
```

Out[25]:
```
(1, 'Python', True, 2.5)
```

In [26]:
```
# type of a tuple
type(tup1)
```

Out[26]:
```
tuple
```

-indexing in tuple

In [27]:
```
tup1[1]
```

Out[27]:
```
'Python'
```

In [28]:
```
tup1[2]
```

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

In [29]:
```
#last element is exclusive
tup1[0:5]
```

Out[29]:
```
(1, 'Python', True, 2.5)
```

In [30]:
```
#count of elements in tuple
len(tup1)
```

Out[30]:
```
4
```

In [31]:
```
tup2 = (2, "Baba Ammar", 3.5, False)
tup2
```

Out[31]:
```
(2, 'Baba Ammar', 3.5, False)
```

In [32]:
```
#concatenate (to add two or more than two tuples)
tup1+tup2
```

Out[32]:
```
(1, 'Python', True, 2.5, 2, 'Baba Ammar', 3.5, False)
```

In [33]:
```
#concatenate + repeat
tup1*2 + tup2
```

Out[33]:
```
(1, 'Python', True, 2.5, 1, 'Python', True, 2.5, 2, 'Baba Ammar', 3.5, False)
```

In [34]:
```
tup3 = (20,50,30,60,79,85)
tup3
```

Out[34]:
```
(20, 50, 30, 60, 79, 85)
```

In [35]:
```
#minimum
min(tup3)
```

Out[35]:
```
20
```

In [36]:
```
#maximum
max(tup3)
```

Out[36]:
```
85
```

In [37]:
```
tup3*2
```

Out[37]:
```
(20, 50, 30, 60, 79, 85, 20, 50, 30, 60, 79, 85)
```

In [38]:
```
tup3.count(30)
```

Out[38]:
```
1
```

In [39]:
```
tup3.index(60)
```

Out[39]:
```
3
```

2- List

- ordered collection of elements
- enclosed in [] square braces/bracket
- Mutateable, you can change the values

In [40]:
```
list1 = [2, "BabaAmmar", False]
list1
```

Out[40]:
```
[2, 'BabaAmmar', False]
```

In [41]:
```
type(list1)
```

Out[41]:
```
list
```

In [42]:
```
len(list1)
```

Out[42]:
```
3
```

In [43]:
```
list1[2]
```

Out[43]:
```
False
```

In [44]:
```
list2 = [3,5,"Ammar", "Codanics", 478, 53.2, False]
list2
```

Out[44]:
```
[3, 5, 'Ammar', 'Codanics', 478, 53.2, False]
```

In [45]:
```
list1 + list2
```

Out[45]:
```
[2, 'BabaAmmar', False, 3, 5, 'Ammar', 'Codanics', 478, 53.2, False]
```

In [46]:
```
list1 *2
```

Out[46]:
```
[2, 'BabaAmmar', False, 2, 'BabaAmmar', False]
```

In [47]:
```
list1.reverse()
list1
```

Out[47]:
```
[False, 'BabaAmmar', 2]
```

In [48]:
```
list1.append("Codanics Youtube channel")
list1
```

```
[False, 'BabaAmmar', 2, 'Codanics Youtube channel']
```

Out[48]:

In [49]:
```
list1.copy()
```

Out[49]:
```
[False, 'BabaAmmar', 2, 'Codanics Youtube channel']
```

In [50]:
```
list1.count(3)
```

Out[50]:
```
0
```

In [51]:
```
list1.extend(list2)
list1
```

Out[51]:
```
[False,
 'BabaAmmar',
 2,
 'Codanics Youtube channel',
 3,
 5,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False]
```

In [52]:
```
list1.index(5)
```

Out[52]:
```
5
```

In [53]:
```
list1.insert(1, "orange")
list1
```

Out[53]:
```
[False,
 'orange',
 'BabaAmmar',
 2,
 'Codanics Youtube channel',
 3,
 5,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False]
```

In [54]:
```
list1.pop(1)
list1
```

Out[54]:
```
[False,
 'BabaAmmar',
 2,
 'Codanics Youtube channel',
 3,
 5,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False]
```

In [55]:
```
list1.remove(5)
list1
```

Out[55]:
```
[False,
 'BabaAmmar',
 2,
 'Codanics Youtube channel',
 3,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False]
```

In [56]:
```
list3 = [20,30,35,50,40,12,15,11,10,356,56,886]
list3
```

Out[56]:
```
[20, 30, 35, 50, 40, 12, 15, 11, 10, 356, 56, 886]
```

In [57]:
```
list3.sort()
```

In [58]:
```
len(list3)
```

Out[58]:
```
12
```

In [59]:
```
#sorting a list
list3.sort()
list3
```

Out[59]:
```
[10, 11, 12, 15, 20, 30, 35, 40, 50, 56, 356, 886]
```

In [60]:
```
list3*3
```

Out[60]:
```
[10,
 11,
 12,
 15,
 20,
 30,
 35,
 40,
 50,
 56,
 356,
 886,
 10,
 11,
 12,
 15,
 20,
 30,
 35,
 40,
 50,
 56,
 356,
 886,
 10,
 11,
 12,
```

```
15,
20,
30,
35,
40,
50,
56,
356,
886]
```

In [61]:
```
lists = list1 + list2
lists
```

Out[61]:
```
[False,
 'BabaAmmar',
 2,
 'Codanics Youtube channel',
 3,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False,
 3,
 5,
 'Ammar',
 'Codanics',
 478,
 53.2,
 False]
```

In [62]:
```
list1.clear()
```

3- Dictionaries

- An unordered collection of elements
- Key and Value
- Curly braces or brackets {}
- Mutateable/Change the values

In [63]:
```
# Food and their prices
food1 = {"Samosa":30, "Pakora":100, "Raita":20, "Salad":50, "Chicken Rolls":30}
food1
```

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

In [64]:
```
type(food1)
```

Out[64]:
```
dict
```

In [65]:
```
#extract data
keys1 = food1.keys()
keys1
```

Out[65]:
```
dict_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Rolls'])
```

In [66]:
```
#extract values
```

```
values1 = food1.values()
values1
```

Out[66]:
```
dict_values([30, 100, 20, 50, 30])
```

In [67]:
```
#adding new element
food1["Tikki"]=10
food1
```

Out[67]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Rolls': 30,
 'Tikki': 10}
```

In [68]:
```
#update the values
food1["Tikki"]=15
food1
```

Out[68]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Rolls': 30,
 'Tikki': 15}
```

In [69]:
```
food2 = {"Dates":50, "Chocolates":200, "Siwayyan":1000}
food2
```

Out[69]:
```
{'Dates': 50, 'Chocolates': 200, 'Siwayyan': 1000}
```

In [70]:
```
#concatenate
food1.update(food2)
food1
```

Out[70]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Rolls': 30,
 'Tikki': 15,
 'Dates': 50,
 'Chocolates': 200,
 'Siwayyan': 1000}
```

In [71]:
```
food1.copy()
```

Out[71]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Rolls': 30,
 'Tikki': 15,
 'Dates': 50,
 'Chocolates': 200,
 'Siwayyan': 1000}
```

In [72]:
```
food1.fromkeys('Samosa', -2)
```

```
{'S': -2, 'a': -2, 'm': -2, 'o': -2, 's': -2}
```

Out[72]:

In [73]:
```
food1.get('Raita')
```

Out[73]:
```
20
```

In [74]:
```
food1.items()
```

Out[74]:
```
dict_items([('Samosa', 30), ('Pakora', 100), ('Raita', 20), ('Salad', 50), ('Chicken Rolls', 30), ('Tikki', 15), ('Dates', 50), ('Chocolates', 200), ('Siwayyan', 1000)])
```

In [75]:
```
food1.popitem()
```

Out[75]:
```
('Siwayyan', 1000)
```

In [76]:
```
food1.values()
```

Out[76]:
```
dict_values([30, 100, 20, 50, 30, 15, 50, 200])
```

In [77]:
```
food1.setdefault('Samosa')
```

Out[77]:
```
30
```

4- Set

- Unordered and unindexed
- curly braces are used {}
- No duplicates allowed

In [78]:
```
s1 = {1,2.2,5.2,"Ammar", "Codanics", "Faisalabad", True}
s1
```

Out[78]:
```
{1, 2.2, 5.2, 'Ammar', 'Codanics', 'Faisalabad'}
```

In [79]:
```
s1.add("Ammar")
s1
```

Out[79]:
```
{1, 2.2, 5.2, 'Ammar', 'Codanics', 'Faisalabad'}
```

In [80]:
```
s1.remove("Ammar")
s1
```

Out[80]:
```
{1, 2.2, 5.2, 'Codanics', 'Faisalabad'}
```

In [81]:
```
s1.copy()
```

Out[81]:
```
{1, 2.2, 5.2, 'Codanics', 'Faisalabad'}
```

In [82]:
```
s2 = s1.copy()
s2
```

Out[82]: {1, 2.2, 5.2, 'Codanics', 'Faisalabad'}

In [83]:
```
s2.add("Nasir")
s2
```

Out[83]: {1, 2.2, 5.2, 'Codanics', 'Faisalabad', 'Nasir'}

In [84]:
```
s2.difference(s1)
```

Out[84]: {'Nasir'}

In [85]:
```
s2.intersection(s1)
```

Out[85]: {1, 2.2, 5.2, 'Codanics', 'Faisalabad'}

In [86]:
```
s2.isdisjoint(s1)
```

Out[86]: False

In [87]:
```
s2.issubset(s1)
```

Out[87]: False

In [88]:
```
s2.issuperset(s1)
```

Out[88]: True

In [89]:
```
s2.pop()
s2
```

Out[89]: {2.2, 5.2, 'Codanics', 'Faisalabad', 'Nasir'}

In [90]:
```
s2.symmetric_difference(s1)
```

Out[90]: {1, 'Nasir'}

In [91]:
```
s2.union(s1)
```

Out[91]: {1, 2.2, 5.2, 'Codanics', 'Faisalabad', 'Nasir'}

In [92]:
```
s2.update(s1)
s2
```

Out[92]: {1, 2.2, 5.2, 'Codanics', 'Faisalabad', 'Nasir'}