Indexing

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

Out[1]: 'Samosa Pakora'

In [2]:
```python
len(a)
```

Out[2]: 13

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

Out[3]: 'S'

In [4]:
```python
a[1]
```

Out[4]: 'a'

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

Out[5]: 'o'

In [6]:
```python
a[12]
```

Out[6]: 'a'

In [7]:
```python
a[0:5]
```

Out[7]: 'Samos'

In [8]:
```python
# Last index exclusive
a[0:13]
```

Out[8]: 'Samosa Pakora'

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

Out[9]: 'r'

In [10]:
```python
a[-6:len(a)]
```

Out[10]:   'Pakora'

In [11]:
```python
food = "biryani"
food
```

Out[11]:   'biryani'

Strings Methods

In [12]:
```python
food
```

Out[12]:   'biryani'

In [13]:
```python
len(food)
```

Out[13]:   7

In [14]:
```python
food.capitalize()
```

Out[14]:   'Biryani'

In [15]:
```python
food
```

Out[15]:   'biryani'

In [16]:
```python
#another method to capitalize first letter
food.title()
```

Out[16]:   'Biryani'

In [17]:
```python
#uppercase Letters
food.upper ( )
```

Out[17]:   'BIRYANI'

In [18]:
```python
#Lowercase Letters
food.lower()
```

Out[18]:   'biryani'

In [19]:
```python
food.replace("b", "Sh")
```

Out[19]:   'Shiryani'

In [20]:

```
# counting a specific aLphabeI in a string
name = "Learning python with baba g"
name
```

Out[20]: 'Learning python with baba g'

In [21]:
```
name.count("a")
```

Out[21]: 3

• finding an index number in string

In [22]:
```
name = "baba _ammar with Dr Ammar Tufail"
name
```

Out[22]: 'baba _ammar with Dr Ammar Tufail'

In [23]:
```
name.find("D")
```

Out[23]: 17

In [24]:
```
# - how to spLit a string
food = "I love samosa, pakora, raita, biryan and karahi"
food
```

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

In [25]:
```
food.split (",")
```

Out[25]: ['I love samosa', ' pakora', ' raita', ' biryan and karahi']

Basic Data Structure in Python

1- Tuple

2- List

3- Dictionaries

4- Set

5- 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 [26]:
```python
tup1 = (1, "Python", True, 2.5)
tup1
```

Out[26]: (1, 'Python', True, 2.5)

In [27]:
```python
# type of a tuple
type(tup1)
```

Out[27]: tuple

-indexing in tuple

In [28]:
```python
tup1[1]
```

Out[28]: 'Python'

In [29]:
```python
tup1[2]
```

Out[29]: True

In [30]:
```python
#last elevent is  exelusive
tup1[0:5]
```

Out[30]: (1, 'Python', True, 2.5)

In [31]:
```python
#count of e Lernents in tuple
len(tup1)
```

Out[31]: 4

In [32]:
```python
tup2 = (2, "Baba Ammar", 3.5, False)
tup2
```

Out[32]: (2, 'Baba Ammar', 3.5, False)

In [33]:
```python
# concatenate (to add two or more than two tup Les)
tup1 + tup2
```

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

In [34]:
```python
# concatenate + repeat
tup1*2 + tup2
```

Out[34]: (1, 'Python', True, 2.5, 1, 'Python', True, 2.5, 2, 'Baba Ammar', 3.5, False)

In [35]:
```python
tup3 = (20,50,30,60,79,85)
tup3
```

Out[35]: (20, 50, 30, 60, 79, 85)

In [36]:
```python
min(tup3)
```

Out[36]: 20

In [37]:
```python
max(tup3)
```

Out[37]: 85

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

Out[38]: 1

In [39]:
```python
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]:
```python
listl = [2, "Arham", False]
listl
```

Out[40]: [2, 'Arham', False]

In [41]:
```python
type(listl)
```

Out[41]: list

In [42]:
```python
len(listl)
```

Out[42]: 3

In [43]:
```python
listl[2]
```

Out[43]: False

In [44]:
```python
list2 = [3,5,"Adeel", "Code", 478, 53.2, False]
list2
```

Out[44]: [3, 5, 'Adeel', 'Code', 478, 53.2, False]

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

Out[45]: [2, 'Arham', False, 3, 5, 'Adeel', 'Code', 478, 53.2, False]

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

Out[46]: [2, 'Arham', False, 2, 'Arham', False]

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

Out[47]: [False, 'Arham', 2]

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

Out[48]: [False, 'Arham', 2, 'Youtube channel']

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

Out[49]: [False, 'Arham', 2, 'Youtube channel']

In [50]:
```python
# .count method return the occurance of the value in a list
```

In [51]:
```python
list1.count(2)
```

Out[51]: 1

In [52]:
```python
list1.extend(list2)
list1
```

Out[52]: [False, 'Arham', 2, 'Youtube channel', 3, 5, 'Adeel', 'Code', 478, 53.2, False]

In [53]:
```python
list1.index(5)
```

Out[53]: 5

In [54]:

```python
listl.insert(1, "orange")
listl
```

Out[54]:
```
[False,
 'orange',
 'Arham',
 2,
 'Youtube channel',
 3,
 5,
 'Adeel',
 'Code',
 478,
 53.2,
 False]
```

In [55]:
```python
listl.pop(1)
listl
```

Out[55]:
```
[False, 'Arham', 2, 'Youtube channel', 3, 5, 'Adeel', 'Code', 478, 53.2, False]
```

In [56]:
```python
listl.remove(5)
listl
```

Out[56]:
```
[False, 'Arham', 2, 'Youtube channel', 3, 'Adeel', 'Code', 478, 53.2, False]
```

In [57]:
```python
list3 = [20,30,35,50,40,12,15,11,10,356,56,886]
list3
```

Out[57]:
```
[20, 30, 35, 50, 40, 12, 15, 11, 10, 356, 56, 886]
```

In [58]:
```python
list3.sort()
```

In [59]:
```python
len(list3)
```

Out[59]:
```
12
```

In [60]:
```python
#sorting a List
list3.sort()
list3
```

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

In [61]:
```python
list3*3
```

Out[61]:
```
[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 [62]:
```python
lists = list1 + list2
lists
```

Out[62]:
```
[False,
 'Arham',
 2,
 'Youtube channel',
 3,
 'Adeel',
 'Code',
 478,
 53.2,
 False,
 3,
 5,
 'Adeel',
 'Code',
 478,
 53.2,
 False]
```

In [ ]:

3- Dictionaries

- An unordered collection of elements

- Key and Value

- Curly braces or brackets ()

- Mutateable/Change the values

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

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

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

Out[64]:   dict

```
In [65]:   #extract data
           keysl = foodl.keys()
           keysl
```

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

```
In [66]:   #extnact vaLues

           valuesl = foodl.values()
           valuesl
```

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

```
In [67]:   #adding new eLernent
           foodl["Tikki"]=10
           foodl
```

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

```
In [68]:   #update the vaLues
           foodl["Tikki"]=15
           foodl
```

Out[68]:   {'Samosa': 30,
            'Pakora': 100,
            'Raita': 20,
            'Salad': 50,
```

```
                   'Chicken Rolls': 30,
                   'Tikki': 15}
```

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

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

In [70]:
```python
#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]:
```python
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]:
```python
food1.fromkeys('Samosa', -2)
```

Out[72]:   `{'S': -2, 'a': -2, 'm': -2, 'o': -2, 's': -2}`

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

Out[73]:   `20`

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

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

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

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

In [76]:
```python
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 ( "Arham" )
s2
```

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

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

Out[84]:   {'Arham'}

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, 'Arham', 'Codanics', 'Faisalabad'}

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

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

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

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

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

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