Indexing

In [1]:

```
# make a string
a="Samosa pakora"
a
```

Out[1]:

```
'Samosa pakora'
```

In [2]:

```
a[1]
```

Out[2]:

```
'a'
```

In [3]:

```
a[3]
```

Out[3]:

```
'o'
```

In [4]:

```
a[12]
```

Out[4]:

```
'a'
```

In [5]:

```
a[0:5]
```

Out[5]:

```
'Samos'
```

In [6]:

```
a[0:6]
```

Out[6]:

```
'Samosa'
```

In [7]:

```
# last Index is exclusive
a[0:12]
```

Out[7]:

```
'Samosa pakor'
```

In [8]:

```
# last Index is exclusive
a[0:13]
```

Out[8]:

```
'Samosa pakora'
```

In [9]:

```
a[-2]
```

Out[9]:

```
'r'
```

In [10]:

```
a[-1]
```

```
Out[10]: 'a'
```

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

```
Out[11]: 'p'
```

```
In [12]: a[-6:13]
```

```
Out[12]: 'pakora'
```

```
In [13]: food="biryani"
         food
```

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

*string methods

```
In [14]: food
```

```
Out[14]: 'biryani'
```

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

```
Out[15]: 7
```

```
In [16]: # capitilize ist beginning element
         food.capitalize()
```

```
Out[16]: 'Biryani'
```

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

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

```
In [18]: # uppercase everry letters
         food.upper()
```

```
Out[18]: 'BIRYANI'
```

```
In [19]: #replace
         food.replace("b","sh")
```

```
Out[19]: 'shiryani'
```

```
In [20]: # counting a specific alphabet in a string
```

```
name="baba_ammar with Dr Aammar Tufail"
name
```

Out[20]:

```
'baba_ammar with Dr Aammar Tufail'
```

In [21]:

```
name.count("a")
```

Out[21]:

```
7
```

finding a index number in string

In [22]:

```
name="baba_ammar with Dr Aammar Tufail"
name
```

Out[22]:

```
'baba_ammar with Dr Aammar Tufail'
```

In [23]:

```
name.find("T")
```

Out[23]:

```
26
```

In [24]:

```
# how to split a strin
food="I love samosa,pakora,biryani,and,karahi"
food
```

Out[24]:

```
'I love samosa,pakora,biryani,and,karahi'
```

In [25]:

```
food.split(",")
```

Out[25]:

```
['I love samosa', 'pakora', 'biryani', '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 parenthesis. *Different kind of elements can be stored.*Once elements are stored you cannot change them(unmutatable)

In [32]:

```
tup1 = (1,"python",True,2.5)
tup1
```

```
Out[32]: (1, 'python', True, 2.5)
```

```
In [34]: # type of a tuple
         type(tup1)
```

```
Out[34]: tuple
```

-Indexing in tuple

```
In [35]: tup1[1]
```

```
Out[35]: 'python'
```

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

```
Out[37]: True
```

```
In [39]: # last element is exclusive
         tup1[0:5]
```

```
Out[39]: (1, 'python', True, 2.5)
```

```
In [40]: # count of elements in tuple
         len(tup1)
```

```
Out[40]: 4
```

```
In [46]: tup2=(2,"baba_Ammar",3.5,False)
         tup2
```

```
Out[46]: (2, 'baba_Ammar', 3.5, False)
```

```
In [47]: # concatimate (to add two tuple or more than)
         tup1+tup2
```

```
Out[47]: (1, 'python', True, 2.5, 2, 'baba_Ammar', 3.5, False)
```

```
In [49]: # concatimate + repeat
         tup1*3+tup2
```

```
Out[49]: (1,
          'python',
          True,
          2.5,
          1,
          'python',
          True,
          2.5,
          1,
          'python',
          True,
          2.5,
          2,
          'baba_Ammar',
          3.5,
          False)
```

In [52]:

```
tup3=(20,50,30,60,79,85)
tup3
```

Out[52]:

```
(20, 50, 30, 60, 79, 85)
```

In [53]:

```
# minimum and max value
min(tup3)
```

Out[53]:

```
20
```

In [54]:

```
max(tup3)
```

Out[54]:

```
85
```

In [55]:

```
tup3*2
```

Out[55]:

```
(20, 50, 30, 60, 79, 85, 20, 50, 30, 60, 79, 85)
```

2. List

- Ordered collection of elements
- Enclosed in []square braces, brackets
- Mutable. you can change the valueb

In [1]:

```
list1=[2,"baba_Ammar",False]
list1
```

Out[1]:

```
[2, 'baba_Ammar', False]
```

In [3]:

```
type (list1)
```

Out[3]:

```
list
```

In [4]:

```
len(list1)
```

Out[4]:

```
3
```

In [6]:

```
list1[2]
```

Out[6]:

```
False
```

In [14]:

```
list2=[3,5,"baba_Ammar","codanics",478,532,False]
list2
```

Out[14]:

```
[3, 5, 'baba_Ammar', 'codanics', 478, 532, False]
```

In [16]:

```
list1+list2
```

Out[16]:

```
[2, 'baba_Ammar', False, 3, 5, 'baba_Ammar', 'codanics', 478, 532, False]
```

In [17]:

```
list1*2
```

Out[17]:

```
[2, 'baba_Ammar', False, 2, 'baba_Ammar', False]
```

In [19]:

```
list2*2
```

Out[19]:

```
[3,
 5,
 'baba_Ammar',
 'codanics',
 478,
 532,
 False,
 3,
 5,
 'baba_Ammar',
 'codanics',
 478,
 532,
 False]
```

In [20]:

```
list1.reverse
list1
```

Out[20]:

```
[2, 'baba_Ammar', False]
```

In [21]:

```
list1
```

Out[21]:

```
[2, 'baba_Ammar', False]
```

In [23]:

```
list1.reverse()
list1
```

Out[23]:

```
[False, 'baba_Ammar', 2]
```

In [27]:

```
list1.append("codanics youtube channel")
```

```
list1
```

Out[27]:

```
[False,
 'baba_Ammar',
 2,
 'codanics youtube channel',
 'codanics youtube channel',
 'codanics youtube channel',
 'codanics youtube channel']
```

In [32]:

```
list3=[20,30,35,50,40,12,10,36,56,89,886]
len(list3)
```

Out[32]:

```
11
```

In [35]:

```
list3.sort()
list3
```

Out[35]:

```
[10, 12, 20, 30, 35, 36, 40, 50, 56, 89, 886]
```

In [36]:

```
list3*2
```

Out[36]:

```
[10,
 12,
 20,
 30,
 35,
 36,
 40,
 50,
 56,
 89,
 886,
 10,
 12,
 20,
 30,
 35,
 36,
 40,
 50,
 56,
 89,
 886]
```

In [37]:

```
list2+list3
```

Out[37]:

```
[3,
 5,
 'baba_Ammar',
 'codanics',
 478,
 532,
 False,
 10,
 12,
 20,
 30,
 35,
 36,
 40,
 50,
 56,
 89,
 886]
```

In [45]:

```
lists=list1+list2+list3
lists
```

Out[45]:

```
[3,
 5,
 'baba_Ammar',
 'codanics',
 478,
 532,
 False,
 3,
 5,
 'baba_Ammar',
 'codanics',
 478,
 532,
 False,
 10,
 12,
 20,
 30,
 35,
 36,
 40,
 50,
 56,
 89,
 886]
```

3. Dictionaries

- An unordered collevtion elements *Key and value*Curly braces or brackets{} *Mutatable or change the value

In [53]:

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

Out[53]:

```
{'Samosa': 30, 'Pakora': 100, 'Raita': 20, 'Salad': 50, 'Chicken Roll': 30}
```

In [47]:

```
type(food1)
```

Out[47]:

```
dict
```

In [54]:

```
# extract data
keys=food1.keys()
keys
```

Out[54]:

```
dict_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Roll'])
```

In [49]:

```

values=food1.values()
values
```

Out[49]:

```
dict_values([30, 100, 20, 50, 30])
```

In [55]:

```

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

Out[55]:

```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 10}
```

In [56]:

```

# update the value
food1["Tikki"]=15
food1
```

Out[56]:

```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 15}
```

In [57]:

```

# update the value
food1["Samosa"]=40
food1
```

Out[57]:

```
{'Samosa': 40,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 15}
```

In [58]:

```

food2={"dates":50,"Chocolates":200,"Swayyan":1000}
food2
```

Out[58]:

```
{'dates': 50, 'Chocolates': 200, 'Swayyan': 1000}
```

In [59]:

```

#Concatinate
food1.update(food2)
```

In [60]:

```
food1
```

Out[60]:

```
{'Samosa': 40,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 15,
 'dates': 50,
 'Chocolates': 200,
 'Swayyan': 1000}
```

4- Set

*Unordered and unindexed*Curly braces are used {} *No duplicates allowed.

In [17]:

```
s1={1,2.2,3.5,"Ammar","Codanics","Faislabad",True}
s1
```

Out[17]:

```
{1, 2.2, 3.5, 'Ammar', 'Codanics', 'Faislabad'}
```

In [19]:

```
s1.add("Ammar1")
s1
```

Out[19]:

```
{1, 2.2, 3.5, 'Ammar', 'Ammar1', 'Codanics', 'Faislabad'}
```

In [20]:

```
s1.remove("Ammar1")
s1
```

Out[20]:

```
{1, 2.2, 3.5, 'Ammar', 'Codanics', 'Faislabad'}
```

In [ ]: