- Indexing

In [6]:
```
a = "samosa or pakora"
a
```

Out[6]:
```
'samosa or pakora'
```

In [10]:
```
# Lenth of string
len(a)
```

Out[10]:
```
16
```

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

Out[7]:
```
'samosa or pakora'
```

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

Out[8]:
```
's'
```

In [9]:
```
a[1]
```

Out[9]:
```
'a'
```

In [11]:
```
a[15]
```

Out[11]:
```
'a'
```

In [12]:
```
a[0:5]
```

Out[12]:
```
'samos'
```

In [13]:
```
a[0:6]
```

Out[13]:
```
'samosa'
```

In [15]:
```
a[0:16]
```

Out[15]:
```
'samosa or pakora'
```

In [32]:
```
a[-6:16]
```

Out[32]:
```
'pakora'
```

String Methods

In [39]:

```
food = "biryani or raita"
food
```

Out[39]:

```
'biryani or raita'
```

In [35]:

```
len(food)
```

Out[35]:

```
7
```

In [40]:

```
# word capitalizing

food.capitalize()
```

Out[40]:

```
'Biryani or raita'
```

In [41]:

```
# convert uppercase

food.upper()
```

Out[41]:

```
'BIRYANI OR RAITA'
```

In [42]:

```
# word replacement

food.replace("b", "sh")
```

Out[42]:

```
'shiryani or raita'
```

In [47]:

```
# cound specific letter in string

name = "hello my name is awais"
name.count("a")
```

Out[47]:

```
3
```

findind index number in string

In [52]:

```
# find index number of specific charctor in string

name.find("w")
```

Out[52]:

```
18
```

In [55]:

```
# split string
```

```
khana = 'i love , samosa, baryani and raita'
khana
```

Out[55]:

```
'i love , samosa, baryani and raita'
```

In [56]:

```
khana.split(",")
```

Out[56]:

```
['i love ', ' samosa', ' baryani and raita']
```

Basic data structure in python

tuple

list

dictnory

set

1-Tuple

. Ordered collection of elements

. enclosed in () braces

. deferent kind of element can stored

. ones element stored u cant b change(unmutatable)

In [5]:

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

tup1
```

Out[5]:

```
(1, 'python', True, 2.5)
```

In [6]:

```
# type of tuple
type(tup1)
```

Out[6]:

```
tuple
```

. indexing in tuple

In [11]:

```
tup1[1]
```

Out[11]:

```
'python'
```

In [10]:

```
tup1[0]
```

```
Out[10]: 1
```

```
In [17]: #last element is exclusive

         tup1[0:2]
```

```
Out[17]: (1, 'python')
```

```
In [18]: len(tup1)
```

```
Out[18]: 4
```

```
In [21]: tup2 = ("azeem aslam", 2.5, False)
         tup2
```

```
Out[21]: ('azeem aslam', 2.5, False)
```

```
In [22]: #concatination

         tup1 + tup2
```

```
Out[22]: (1, 'python', True, 2.5, 'azeem aslam', 2.5, False)
```

```
In [25]: #concationation and repitation

         tup1*2 + tup2
```

```
Out[25]: (1, 'python', True, 2.5, 1, 'python', True, 2.5, 'azeem aslam', 2.5, False)
```

```
In [26]: tup3 = (10, 20, 30, 80, 50)
         tup3
```

```
Out[26]: (10, 20, 30, 80, 50)
```

```
In [27]: min(tup3)
```

```
Out[27]: 10
```

```
In [28]: max(tup3)
```

```
Out[28]: 80
```

```
In [29]: tup3*2
```

```
(10, 20, 30, 80, 50, 10, 20, 30, 80, 50)
```

Out[29]:

---

2 - List

. oedered collection of elements

. enclosed in [] square brackets

. mutateabale u can change values

In [30]:
```
list1 = [1, "azeem list", 2.5]
list1
```

Out[30]:
```
[1, 'azeem list', 2.5]
```

In [31]:
```
type(list1)
```

Out[31]:
```
list
```

In [32]:
```
len(list1)
```

Out[32]:
```
3
```

In [35]:
```
list1[2]
```

Out[35]:
```
2.5
```

In [36]:
```
list2 = [4, 5, "azeem", False, 455, 657]
list2
```

Out[36]:
```
[4, 5, 'azeem', False, 455, 657]
```

In [37]:
```
list1 + list2
```

Out[37]:
```
[1, 'azeem list', 2.5, 4, 5, 'azeem', False, 455, 657]
```

In [38]:
```
list1*2
```

Out[38]:
```
[1, 'azeem list', 2.5, 1, 'azeem list', 2.5]
```

In [42]:
```
list1.reverse()
list1
```

Out[42]:
```
[2.5, 'azeem list', 1]
```

In [44]:

```
list1.append("ye jor dia list1 k sath")
list1
```

Out[44]:

```
[2.5, 'azeem list', 1, 'ye jor dia list1 k sath', 'ye jor dia list1 k sath']
```

In [56]:

```
list3 = [20,90,40,500,30,10]
list3
```

Out[56]:

```
[20, 90, 40, 500, 30, 10]
```

In [57]:

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

Out[57]:

```
[10, 20, 30, 40, 90, 500]
```

In [55]:

```
# Method        Description
# append()      Adds an element at the end of the list
# clear()       Removes all the elements from the list
# copy()        Returns a copy of the list
# count()       Returns the number of elements with the specified value
# extend()      Add the elements of a list (or any iterable), to the end of the current
# index()       Returns the index of the first element with the specified value
# insert()      Adds an element at the specified position
# pop() Removes the element at the specified position
# remove()      Removes the item with the specified value
# reverse()     Reverses the order of the list
# sort()        Sorts the list
```

3 -Dictioneries

. unordered collection of element

. key and value pair

. use with curly {} braces

. mutateabale

In [61]:

```
food1 = {"samosa" :20, "pakora" :30, "raita" :10}
food1
```

Out[61]:

```
{'samosa': 20, 'pakora': 30, 'raita': 10}
```

In [62]:

```
type(food1)
```

Out[62]:

```
dict
```

In [64]:

```
# extract data   seperate key and value
```

```
keys1 = food1.keys()
keys1
```

Out[64]:

```
dict_keys(['samosa', 'pakora', 'raita'])
```

In [66]:

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

Out[66]:

```
dict_values([20, 30, 10])
```

In [67]:

```
#adding new element

food1["tikki"]=12
food1
```

Out[67]:

```
{'samosa': 20, 'pakora': 30, 'raita': 10, 'tikki': 12}
```

In [68]:

```
#updating values
food1["tikki"]=15
food1
```

Out[68]:

```
{'samosa': 20, 'pakora': 30, 'raita': 10, 'tikki': 15}
```

In [70]:

```
food2 = {"khajor":50, "chocholate":300, "savyan": 500}
food2
```

Out[70]:

```
{'khajor': 50, 'chocholate': 300, 'savyan': 500}
```

In [77]:

```
#concatination of dict use uptade method

food1.update(food2)
food1
```

Out[77]:

```
{'samosa': 20,
 'pakora': 30,
 'raita': 10,
 'tikki': 15,
 'khajor': 50,
 'chocholate': 300,
 'savyan': 500}
```

3 - Sets

. unordered list of eliment

. use {} braces

. no duplication allow

In [94]:

```
s1 = {1, 2, 3, 4.3, "awaisaslam", "awais",  "pattoki"}
s1
```

Out[94]:

```
{1, 2, 3, 4.3, 'awais', 'awaisaslam', 'pattoki'}
```

In [95]:

```
s1.add("aslam")
s1
```

Out[95]:

```
{1, 2, 3, 4.3, 'aslam', 'awais', 'awaisaslam', 'pattoki'}
```

In [ ]:

In [96]:

```
s1.remove("awais")
s1
```

Out[96]:

```
{1, 2, 3, 4.3, 'aslam', 'awaisaslam', 'pattoki'}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: