In [127…
```
a="samosa pakora"
a
```
Out[127…
```
'samosa pakora'
```

In [128…
```
a[0]
```
Out[128…
```
's'
```

In [129…
```
a[5]
```
Out[129…
```
'a'
```

In [130…
```
a[0:6]
```
Out[130…
```
'samosa'
```

In [131…
```
# last index is exclusive
a[0:13]
```
Out[131…
```
'samosa pakora'
```

In [132…
```
a[-2]
```
Out[132…
```
'r'
```

In [133…
```
a[-5]
```
Out[133…
```
'a'
```

In [134…
```
len(a)
```
Out[134…
```
13
```

In [135…
```
a[-6:0]
```
Out[135…
```
''
```

In [136…
```
a[-6:-1]
```
Out[136…
```
'pakor'
```

In [137…
```
a[-5:5]
```

Out[137…
```
''
```

In [138…
```
a[-13:-1]
```

Out[138…
```
'samosa pakor'
```

Strings Method

In [139…
```
food="beryani"
food
```

Out[139…
```
'beryani'
```

In [140…
```
len(food)
```

Out[140…
```
7
```

In [141…
```
food.capitalize()
```

Out[141…
```
'Beryani'
```

In [142…
```
food.upper()
```

Out[142…
```
'BERYANI'
```

In [143…
```
food.lower()
```

Out[143…
```
'beryani'
```

In [144…
```
# replace
food.replace("b","sh")
```

Out[144…
```
'sheryani'
```

In [145…
```
# counting a specific alphabet in a string
name="baba_aamar with salman"
name
```

Out[145…
```
'baba_aamar with salman'
```

In [146…
```
name.count("ba")
```

Out[146…
```
2
```

In [147…
```
name.count("a")
```

Out[147… 7

Finding an index number in a string

In [148…
```
name="baba_aamar with salman_khan"
name
```

Out[148… 'baba_aamar with salman_khan'

In [149…
```
name.find("s")
```

Out[149… 16

In [150…
```
name.find("aa")
```

Out[150… 5

In [151…
```
name.find("kh")
```

Out[151… 23

How to split a string

In [152…
```
food="I love samosa, pakora, raita, beryani and karahi"
food
```

Out[152… 'I love samosa, pakora, raita, beryani and karahi'

In [153…
```
food.split(",")
```

Out[153… ['I love samosa', ' pakora', ' raita', ' beryani and karahi']

Basic data structure in python

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

Tuple

- Ordered collection of elements
- Enclosed in parenthesis ()
- Different kinds of elements can be stored

- Once element are stored you can not change them
- Tuple. Tuples are used to store multiple items in a single variable.
- Tuple is one of 4 built-in data types in Python used to store collections of data.
- The other 3 are List, Set, and Dictionary, all with different qualities and usage.

In [154…
```
tup1=(1,"python",True,2.5)
tup1
```

Out[154…
```
(1, 'python', True, 2.5)
```

In [155…
```
# type of tuple
type(tup1)
```

Out[155…
```
tuple
```

Indexing in tuple

In [156…
```
tup1[1]
```

Out[156…
```
'python'
```

In [157…
```
tup1[0]
```

Out[157…
```
1
```

In [158…
```
tup1[2]
```

Out[158…
```
True
```

In [159…
```
tup1[0:3]
```

Out[159…
```
(1, 'python', True)
```

In [160…
```
tup1[0:4]
```

Out[160…
```
(1, 'python', True, 2.5)
```

In [161…
```
# count element in tuple
len(tup1)
```

Out[161…
```
4
```

In [162…
```
tup2=(2,"baba aamar",False,5)
tup2
```

Out[162… (2, 'baba aamar', False, 5)

In [163…
```
# Cancatinate ( to add two or more tuples)

tup1+tup2
```

Out[163… (1, 'python', True, 2.5, 2, 'baba aamar', False, 5)

In [164…
```
# Cancatinate and repeat them ( power is used to repeat it that much times)
tup1*2+tup2
```

Out[164… (1, 'python', True, 2.5, 1, 'python', True, 2.5, 2, 'baba aamar', False, 5)

In [165…
```
tup3=(40,50,60,70,80)
tup3
```

Out[165… (40, 50, 60, 70, 80)

In [166…
```
# minimum
min(tup3)
```

Out[166… 40

In [167…
```
# maximum
max(tup3)
```

Out[167… 80

In [168…
```
tup3*3
```

Out[168… (40, 50, 60, 70, 80, 40, 50, 60, 70, 80, 40, 50, 60, 70, 80)

In [169…
```
tup3*5
```

Out[169…
```
(40,
 50,
 60,
 70,
 80,
 40,
 50,
 60,
 70,
 80,
 40,
 50,
 60,
 70,
 80,
 40,
```

```
50,
60,
70,
80,
40,
50,
60,
70,
80)
```

List

- ordered collection of elements
- Enclosed in square brackets []
- Mutateable, means you can change the values

In [170…
```
list1 = [2, "baba_aamar", False]
list1
```

Out[170…
```
[2, 'baba_aamar', False]
```

In [171…
```
type(list1)
```

Out[171…
```
list
```

In [172…
```
list1[1]
```

Out[172…
```
'baba_aamar'
```

In [173…
```
list1[2]
```

Out[173…
```
False
```

In [174…
```
list1[0:2]
```

Out[174…
```
[2, 'baba_aamar']
```

In [175…
```
list2 = [2.5, 7, "salman_khan", 8.5,True]
list2
```

Out[175…
```
[2.5, 7, 'salman_khan', 8.5, True]
```

In [176…
```
list1+list2
```

Out[176…
```
[2, 'baba_aamar', False, 2.5, 7, 'salman_khan', 8.5, True]
```

In [177…
```
list1*2+list2
```

Out[177…
```
[2,
 'baba_aamar',
 False,
 2,
 'baba_aamar',
 False,
 2.5,
 7,
 'salman_khan',
 8.5,
 True]
```

In [178…
```
list1
```

Out[178…
```
[2, 'baba_aamar', False]
```

In [179…
```
list1.reverse()
list1
```

Out[179…
```
[False, 'baba_aamar', 2]
```

In [180…
```
list1.append("my name is khan")
list1

# every time you execute it, it will add the new element into list1
```

Out[180…
```
[False, 'baba_aamar', 2, 'my name is khan']
```

List.count is an Assignment, Also practice on other fuctions of list

In [181…
```
list1.count("my")
```

Out[181…
```
0
```

In [182…
```
list1.count("m")
```

Out[182…
```
0
```

In [183…
```
list3 = [50,10,89,48,94,77,34]
list3
```

Out[183…
```
[50, 10, 89, 48, 94, 77, 34]
```

In [184…
```
list3.sort()
list3
```

Out[184…
```
[10, 34, 48, 50, 77, 89, 94]
```

In [185…
```
list1+list2+list3
```

Out[185…
```
[False,
 'baba_aamar',
 2,
 'my name is khan',
 2.5,
 7,
 'salman_khan',
 8.5,
 True,
 10,
 34,
 48,
 50,
 77,
 89,
 94]
```

Dictionaries

- An unordered collection of elements
- Key and value
- Braces {}
- Mutateable

In [186…
```
# Food and their prices
# the first element is key whereas the second element is value.

food1 = {"samosa":30, "pakora":20, "raita":15, "salad":10, "chicken rolls": 50}
food1
```

Out[186…
```
{'samosa': 30, 'pakora': 20, 'raita': 15, 'salad': 10, 'chicken rolls': 50}
```

In [187…
```
type(food1)
```

Out[187…
```
dict
```

In [188…
```
# Extract data

keys1= food1.keys()
keys1
```

Out[188…
```
dict_keys(['samosa', 'pakora', 'raita', 'salad', 'chicken rolls'])
```

In [189…
```
values1 = food1.values()
values1
```

Out[189…
```
dict_values([30, 20, 15, 10, 50])
```

In [190…

```
# Adding new element
food1["tikka"]= 70
food1
```

Out[190…
```
{'samosa': 30,
 'pakora': 20,
 'raita': 15,
 'salad': 10,
 'chicken rolls': 50,
 'tikka': 70}
```

In [191…
```
food2= {"dates":25, "choclates":15, "milk":50}
food2
```

Out[191…
```
{'dates': 25, 'choclates': 15, 'milk': 50}
```

In [192…
```
# Cancatinate
food1.update(food2)
food1
```

Out[192…
```
{'samosa': 30,
 'pakora': 20,
 'raita': 15,
 'salad': 10,
 'chicken rolls': 50,
 'tikka': 70,
 'dates': 25,
 'choclates': 15,
 'milk': 50}
```

4- Sets

- Unordered and unindexed
- braces are used
- No duplicate allowed

In [193…
```
s1= {2.4, 15,"aamarr", "salman",78, True}
s1
```

Out[193…
```
{15, 2.4, 78, True, 'aamarr', 'salman'}
```

In [194…
```
type(s1)
```

Out[194…
```
set
```

In [195…
```
s1.add("khan")
s1
```

Out[195…
```
{15, 2.4, 78, True, 'aamarr', 'khan', 'salman'}
```

In [196…
```
s1.remove("khan")
```

```
s1
```

Out[196…
```
{15, 2.4, 78, True, 'aamarr', 'salman'}
```

In [197…
```
# no duplication of elements happens
s1.add("salman")
s1
```

Out[197…
```
{15, 2.4, 78, True, 'aamarr', 'salman'}
```

In [ ]:
```

```