-Indexing

In [2]:

```
# Make a string
a = "Samosa Pakora"
a
```

Out[2]:

```
'Samosa Pakora'
```

In [3]:

```
a
```

Out[3]:

```
'Samosa Pakora'
```

In [4]:

```
a[0]
```

Out[4]:

```
'S'
```

In [5]:

```
a[6]
```

Out[5]:

```
' '
```

In [6]:

```
# length of indices
len(a)
```

Out[6]:

```
13
```

In [7]:

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

Out[7]:

```
'Samos'
```

In [8]:

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

Out[8]:

```
'Samosa'
```

In [9]:

```
a[0:12]
```

Out[9]:

```
'Samosa Pakor'
```

In [10]:

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

Out[10]:

```
'Samosa Pakora'
```

In [11]:

```
a[-1]
```

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

```
In [12]: a[-2]
```

```
Out[12]: 'r'
```

```
In [13]: a[-1:-6]
```

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

```
In [14]: a[-6:-1]
```

```
Out[14]: 'Pakor'
```

```
In [29]: a[-1:-1]
```

```
Out[29]: 'Samosa Pakor'
```

```
In [16]: a[-2:-1]
```

```
Out[16]: 'r'
```

```
In [17]: a[-3:-1]
```

```
Out[17]: 'or'
```

```
In [18]: a[-4:-1]
```

```
Out[18]: 'kor'
```

```
In [19]: a[-6]
```

```
Out[19]: 'P'
```

```
In [20]: a[-6:-5]
```

```
Out[20]: 'P'
```

```
In [21]: a[-6:5]
```

```
Out[21]: ''
```

```
In [22]: a[-6:13]
```

```
Out[22]: 'Pakora'
```

In [23]:

```
a[-6:1]
```

```
Out[23]: ''
```

In [37]:

```
a[-7:-6]
```

```
Out[37]: ' '
```

In [38]:

```
a[-1:-1]
```

```
Out[38]: ''
```

In [39]:

```
a[-1:-6]
```

```
Out[39]: ''
```

In [40]:

```
a[-6:-1]
```

```
Out[40]: 'Pakor'
```

In [41]:

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

```
Out[41]: 'Pakora'
```

In [42]:

```
a[-8:-1]
```

```
Out[42]: 'a Pakor'
```

In [44]:

```
food = "biryani"
food
```

```
Out[44]: 'biryani'
```

String Method

In [48]:

```
food
```

```
Out[48]: 'biryani'
```

In [49]:

```
len(food)
```

```
Out[49]: 7
```

In [51]:

```
food.capitalize()
```

Out[51]:

```
'Biryani'
```

In [52]:

```
food.upper()
```

Out[52]:

```
'BIRYANI'
```

In [53]:

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

Out[53]:

```
'shiryani'
```

In [54]:

```
# counting a specific alphabet in a string
name = "baba_aamar with Dr.Aamar Tufail"
name
```

Out[54]:

```
'baba_aamar with Dr.Aamar Tufail'
```

In [55]:

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

Out[55]:

```
8
```

In [56]:

```
name.count("b")
```

Out[56]:

```
2
```

In [57]:

```
name.count("A")
```

Out[57]:

```
1
```

- finding an index number in string

In [58]:

```
name = "baba_aamar with Dr.Aamar Tufail"
name
```

Out[58]:

```
'baba_aamar with Dr.Aamar Tufail'
```

In [59]:

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

Out[59]:

```
25
```

In [62]:

```
name.find("aa")
```

Out[62]:

```
5
```

In [63]:
```
name.find("D")
```

Out[63]:
```
16
```

- how to split a string

In [64]:
```
food = "I love samosa, pakora, raita, biryani and karahi"
food
```

Out[64]:
```
'I love samosa, pakora, raita, biryani and karahi'
```

In [65]:
```
food.split(",")
```

Out[65]:
```
['I love samosa', ' pakora', ' raita', ' biryani and karahi']
```

Basic data structures in Python

1-Tuple

2-List

3-Dictionaries

4-Set

1-Tuple

----ordered collection of elements

----enclosed in round brackets ()

----diff kind of elements(int,float,string,boolean) can be stored

----stored element cannot be changed

In [68]:
```
## 1-Tuple
tup1 = (1, "Python", True, 2.5)
tup1
```

Out[68]:
```
(1, 'Python', True, 2.5)
```

In [69]:
```
# type of tuple
type(tup1)
```

```
Out[69]: tuple
```

- Indexing in tuple

```
In [71]: tup1[1]

Out[71]: 'Python'

In [72]: tup1[0]

Out[72]: 1

In [73]: tup1[2]

Out[73]: True

In [74]: tup1[3]

Out[74]: 2.5

In [75]: tup1[0:4]

Out[75]: (1, 'Python', True, 2.5)

In [76]: tup1[0:3]

Out[76]: (1, 'Python', True)

In [77]: tup1[-1]

Out[77]: 2.5

In [78]: tup1[-4:-1]

Out[78]: (1, 'Python', True)

In [79]: tup1[-4:4]

Out[79]: (1, 'Python', True, 2.5)

In [80]: tup1[-4:5]

Out[80]: (1, 'Python', True, 2.5)
```

In [81]:

```
len(tup1)
```

Out[81]:

```
4
```

In [82]:

```
tup2 = (2,"baba aamar", 3.5, False)
tup2
```

Out[82]:

```
(2, 'baba aamar', 3.5, False)
```

In [83]:

```
#concatenate---to add tuple
tup1+tup2
```

Out[83]:

```
(1, 'Python', True, 2.5, 2, 'baba aamar', 3.5, False)
```

In [85]:

```
#concatenate + repeat
tup1*2+tup2
```

Out[85]:

```
(1, 'Python', True, 2.5, 1, 'Python', True, 2.5, 2, 'baba aamar', 3.5, False)
```

In [86]:

```
tup1*2+tup2*2
```

Out[86]:

```
(1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 2,
 'baba aamar',
 3.5,
 False,
 2,
 'baba aamar',
 3.5,
 False)
```

In [89]:

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

Out[89]:

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

In [ ]:

In [90]:

```
min(tup3)
```

Out[90]:

```
20
```

In [91]:

```
max(tup3)
```

Out[91]:

```
60
```

In [92]:

```
tup3*3
```

Out[92]:

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

In [243…

```
# The count() method returns the number of times the specified element appears in the t
tup3.count(50)
tup3
```

Out[243…

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

In [247…

```
tup2.count(3.5)
tup2
```

Out[247…

```
(2, 'baba aamar', 3.5, False)
```

In [254…

```
count1 = tup3.count(20)
count1
```

Out[254…

```
1
```

In [256…

```
count1 = tup2.count(False)
count1
```

Out[256…

```
1
```

In [257…

```
# The index() method returns the index of the specified element in the tuple.
index1 = tup3.index(30)
index1
```

Out[257…

```
1
```

In [ ]:

In [ ]:

2-List

- ordered collection of elements -enclosed in square [] brackets -elements can be changed(mutable)

In [95]:

```
list1 = [1, "Faizan", False]
```

```
list1
```

Out[95]:
```
[1, 'Faizan', False]
```

In [259…
```
#clear all elements of a list
list1.clear()
list1
```

Out[259…
```
[]
```

In [263…
```
list1.append(1)
list1
```

Out[263…
```
[1]
```

In [264…
```
list1.append("faizan")
list1
```

Out[264…
```
[1, 'faizan']
```

In [265…
```
list1.append(False)
list1
```

Out[265…
```
[1, 'faizan', False]
```

In [266…
```
# insert funct----inserts a given element at a given index in a list.
list1.insert(0, False)
list1
```

Out[266…
```
[False, 1, 'faizan', False]
```

In [269…
```
list1.copy()
```

Out[269…
```
[False, 1, 'faizan', False]
```

In [272…
```
copy_list1 = list1.copy()
copy_list1
```

Out[272…
```
[False, 1, 'faizan', False]
```

In [275…
```
# extend------adds the specified list elements (or any iterable) to the end of the curr
list1.extend("faizan")
list1
```

Out[275…
```
[False, 1, 'faizan', False, 'f', 'a', 'i', 'z', 'a', 'n']
```

In [278…
```
list1.extend("goooo")
```

```
list1
```

Out[278…

```
[False,
 1,
 'faizan',
 False,
 'f',
 'a',
 'i',
 'z',
 'a',
 'n',
 'F',
 'a',
 'l',
 's',
 'e',
 'g',
 'o',
 'o',
 'o',
 'o']
```

In [283…

```
len(list1)
```

Out[283…

```
18
```

In [284…

```
list1
```

Out[284…

```
[False,
 1,
 False,
 'f',
 'a',
 'i',
 'z',
 'a',
 'n',
 'F',
 'a',
 'l',
 's',
 'e',
 'g',
 'o',
 'o',
 'o']
```

In [287…

```
#reverse--- reverse order of elements of a list
list3.reverse()
list3
```

Out[287…

```
[100, 99, 67, 65, 34, 33, 11, 1]
```

In [98]:

```
type(list1)
```

Out[98]:
```
list
```

In [99]:
```
len(list1)
```

Out[99]:
```
3
```

In [100…
```
list1[2]
```

Out[100…
```
False
```

In [101…
```
list2 = [3, 5, "aamar", 478, 53.2, False]
list2
```

Out[101…
```
[3, 5, 'aamar', 478, 53.2, False]
```

In [102…
```
len(list2)
```

Out[102…
```
6
```

In [103…
```
list1+list2
```

Out[103…
```
[1, 'Faizan', False, 3, 5, 'aamar', 478, 53.2, False]
```

In [106…
```
list1.reverse()
list1
```

Out[106…
```
[False, 'Faizan', 1]
```

In [109…
```
list1.append("jfkhlkfjslfhs")
list1
```

Out[109…
```
[False, 'Faizan', 1, 'jfkhlkfjslfhs', 'jfkhlkfjslfhs', 'jfkhlkfjslfhs']
```

In [110…
```
list1.append("a")
list1
```

Out[110…
```
[False, 'Faizan', 1, 'jfkhlkfjslfhs', 'jfkhlkfjslfhs', 'jfkhlkfjslfhs', 'a']
```

In [111…
```
list1.remove("jfkhlkfjslfhs")
list1
```

Out[111…
```
[False, 'Faizan', 1, 'jfkhlkfjslfhs', 'jfkhlkfjslfhs', 'a']
```

In [112…
```
list1.remove("jfkhlkfjslfhs")
list1
```

Out[112…

```
[False, 'Faizan', 1, 'jfkhlkfjslfhs', 'a']
```

In [113…

```
list1.remove("jfkhlkfjslfhs")
list1
```

Out[113…

```
[False, 'Faizan', 1, 'a']
```

In [114…

```
list1.remove("a")
list1
```

Out[114…

```
[False, 'Faizan', 1]
```

In [118…

```
list1.append("Faizan")
list1
```

Out[118…

```
[False, 'Faizan', 1, 'Faizan', 'Faizan', 'Faizan', 'Faizan']
```

In [122…

```
list1.pop(1)
list1
```

Out[122…

```
[False, 'Faizan', 'Faizan', 'Faizan']
```

In [123…

```
len(list1)
```

Out[123…

```
4
```

In [124…

```
list1.append(1)
list1
```

Out[124…

```
[False, 'Faizan', 'Faizan', 'Faizan', 1]
```

In [125…

```
list1.pop
list1
```

Out[125…

```
[False, 'Faizan', 'Faizan', 'Faizan', 1]
```

In [126…

```
list1.pop(1)
list1
```

Out[126…

```
[False, 'Faizan', 'Faizan', 1]
```

In [127…

```
list1.pop(1)
list1
```

Out[127…

```
[False, 'Faizan', 1]
```

In [128…

```
list1.pop(1)
list1
```

Out[128…

```
[False, 1]
```

In [129…

```
list1.pop(1)
list1
```

Out[129…

```
[False]
```

In [131…

```
len(list1)
```

Out[131…

```
1
```

In [132…

```
list3 = [33, 34, 65, 67]
list3
```

Out[132…

```
[33, 34, 65, 67]
```

In [134…

```
list3.append(99)
list3
```

Out[134…

```
[33, 34, 65, 67, 99]
```

In [136…

```
list3.append( 100)
list3
```

Out[136…

```
[33, 34, 65, 67, 99, 100]
```

In [137…

```
list3.append(11)
list3
```

Out[137…

```
[33, 34, 65, 67, 99, 100, 11]
```

In [138…

```
list3.append(1)
list3
```

Out[138…

```
[33, 34, 65, 67, 99, 100, 11, 1]
```

In [139…

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

Out[139…

```
[1, 11, 33, 34, 65, 67, 99, 100]
```

In [140…

```
list3*3
```

```
[1,
```

Out[140…

```
 11,
 33,
 34,
 65,
 67,
 99,
 100,
 1,
 11,
 33,
 34,
 65,
 67,
 99,
 100,
 1,
 11,
 33,
 34,
 65,
 67,
 99,
 100]
```

In [141…

```
list1+list2
```

Out[141…

```
[False, 3, 5, 'aamar', 478, 53.2, False]
```

In [142…

```
list1+list3
```

Out[142…

```
[False, 1, 11, 33, 34, 65, 67, 99, 100]
```

In [144…

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

Out[144…

```
[False, 3, 5, 'aamar', 478, 53.2, False, 1, 11, 33, 34, 65, 67, 99, 100]
```

3-Dictionaries

collection of un-ordered elements

key & value

curly braces used{}

mutable-- values can be changed

In [148…

```
#Food and their prices
menu1 = {"dozen samosa":120,"Pakora/kg":200,"Raita":10,"Salad":20,"Chicken roll/piece":
menu1
```

Out[148…

```
{'dozen samosa': 120,
 'Pakora/kg': 200,
 'Raita': 10,
 'Salad': 20,
 'Chicken roll/piece': 80}
```

In [296…

```
keys = {"dozen samosa", "Pakora/kg", "Raita", "Salad", "Chicken roll/piece"}
values = "10% discount"
new = dict.fromkeys(keys,values)
new
```

Out[296…

```
{'dozen samosa': '10% discount',
 'Pakora/kg': '10% discount',
 'Salad': '10% discount',
 'Chicken roll/piece': '10% discount',
 'Raita': '10% discount'}
```

In [299…

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964}

x = car.get("model")
x
```

Out[299…

```
'Mustang'
```

In [300…

```
#get----retrieve a key for specified value
x = menu1.get("dozen samosa")
x
```

Out[300…

```
120
```

In [301…

```
menu1.items()
menu1
```

Out[301…

```
{'dozen samosa': 120,
 'Pakora/kg': 200,
 'Raita': 10,
 'Salad': 20,
 'Chicken roll/piece': 80,
 'Dates': 50,
 'chocolates': 200,
 'swayyan': 1000}
```

In [302…

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

car.popitem()

print(car)
```

```
{'brand': 'Ford', 'model': 'Mustang'}
```

In [303…

```
#popitem---removes last item
menu1.popitem()
menu1
```

Out[303…

```
{'dozen samosa': 120,
 'Pakora/kg': 200,
 'Raita': 10,
 'Salad': 20,
 'Chicken roll/piece': 80,
 'Dates': 50,
 'chocolates': 200}
```

In [149…

```
type(menu1)
```

Out[149…

```
dict
```

In [152…

```
# extract dict data
keys1 = menu1.keys()
keys1
```

Out[152…

```
dict_keys(['dozen samosa', 'Pakora/kg', 'Raita', 'Salad', 'Chicken roll/piece'])
```

In [153…

```
values1 = menu1.values()
values1
```

Out[153…

```
dict_values([120, 200, 10, 20, 80])
```

In [235…

```
#adding new element
menu1.update["tikki"]= 10
menu1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\GEOLOG~1\AppData\Local\Temp/ipykernel_9880/3297672777.py in <module>
      1 #adding new element
----> 2 menu1.update["tikki"]= 10
      3 menu1

TypeError: 'builtin_function_or_method' object does not support item assignment
```

In [236…

```
menu2 = {"Dates":50, "chocolates":200, "swayyan":1000}
menu2
```

Out[236…

```
{'Dates': 50, 'chocolates': 200, 'swayyan': 1000}
```

In [237…

```
# concatenate
menu1.update(menu2)
menu1
```

```
{'dozen samosa': 120,
```

Out[237…

```
 'Pakora/kg': 200,
 'Raita': 10,
 'Salad': 20,
 'Chicken roll/piece': 80,
 'Dates': 50,
 'chocolates': 200,
 'swayyan': 1000}
```

4-Set

un-ordered and un-indexed collection of elements

use curly{} braces

no duplicates allowed

In [239…

```
s1 = {1, 2.2, 5.2, "Aamar", "Codanics", "Faisalabad", True}
s1
```

Out[239…

```
{1, 2.2, 5.2, 'Aamar', 'Codanics', 'Faisalabad'}
```

In [240…

```
s1.add("Aamar")
s1
```

Out[240…

```
{1, 2.2, 5.2, 'Aamar', 'Codanics', 'Faisalabad'}
```

In [241…

```
s1.add("Aamar1")
s1
```

Out[241…

```
{1, 2.2, 5.2, 'Aamar', 'Aamar1', 'Codanics', 'Faisalabad'}
```

In [242…

```
s1.remove("Aamar1")
s1
```

Out[242…

```
{1, 2.2, 5.2, 'Aamar', 'Codanics', 'Faisalabad'}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: