- indexing

In [1]:
```
#make a string
a ="pakora samosa"
a
```

Out[1]:
```
'pakora samosa'
```

In [2]:
```
#lenth of indices
len (a)
```

Out[2]:
```
13
```

In [3]:
```
a[0:5]
```

Out[3]:
```
'pakor'
```

In [ ]:
```
#Last index is exclusive
```

In [4]:
```
a[0:13]
```

Out[4]:
```
'pakora samosa'
```

In [5]:
```
a[-6:-1]
```

Out[5]:
```
'samos'
```

In [6]:
```
a[-4]
```

Out[6]:
```
'm'
```

String Method

In [7]:
```
food = ("biryani")
food
```

Out[7]:
```
'biryani'
```

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

In [8]:
```
#Capitalize every element
food.capitalize()
```

```
Out[8]: 'Biryani'
```

```
In [9]:
#Upper Case Letters
food.upper()
```

```
Out[9]: 'BIRYANI'
```

```
In [10]:
#lower case letters
food.lower()
```

```
Out[10]: 'biryani'
```

```
In [11]:
#replace
food.replace("b", "sh")
```

```
Out[11]: 'shiryani'
```

```
In [12]:
#count a specific letter in a string
name=("baba_ammar_with_dr_ammar")
name
```

```
Out[12]: 'baba_ammar_with_dr_ammar'
```

```
In [13]:
name.count("a")
```

```
Out[13]: 6
```

-finding an index number in string

```
In [14]:
name=("baba_ammar_with_dr_ammar")
name.find("r")
```

```
Out[14]: 9
```

-how to split a string

```
In [15]:
food = ("I love samosa, pakora, biryani and raita")
food
```

```
Out[15]: 'I love samosa, pakora, biryani and raita'
```

```
In [16]:
food.split(",")
```

```
Out[16]: ['I love samosa', ' pakora', ' biryani and raita']
```

-Basic data structure in python

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

-Tuple

- ordered collection of elements
- enclosed in round () braces / paranthesis
- different type of elements can be stored
- Once elements are stored you can not change them (unmutable)

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

```
tuple
```

In [19]:

```
tup2 = (2, "example", False, 2.5)
tup2
```

Out[19]:

```
(2, 'example', False, 2.5)
```

In [20]:

```
type(tup2)
```

Out[20]:

```
tuple
```

In [21]:

```
len(tup2)
```

Out[21]:

```
4
```

-indexing in tuple

In [22]:

```
tup1[1]
```

Out[22]:

```
'python'
```

In [23]:

```
tup1[2]
```

Out[23]:

```
True
```

In [ ]:

```
#last element is exclusive
```

In [24]:

```
len(tup1)
```

Out[24]:

```
4
```

In [25]:

```
tup1[0:4]
```

Out[25]:

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

In [26]:

```
tup1[0]
```

Out[26]:

```
1
```

In [27]:

```
tup1 + tup2
```

Out[27]:

```
(1, 'python', True, 2.5, 2, 'example', False, 2.5)
```

In [28]:

```
tup2 = (2, "babaamaar", 3.5, False)
tup2
```

Out[28]:

```
(2, 'babaamaar', 3.5, False)
```

In [29]:

```
tup1+tup2 #concatinate
```

Out[29]:

```
(1, 'python', True, 2.5, 2, 'babaamaar', 3.5, False)
```

In [30]:

```
tup1*3 + tup1
```

Out[30]:

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

In [31]:

```
tup3 = (10, 20, 30, 40, 50, 65)
tup3
```

Out[31]:

```
(10, 20, 30, 40, 50, 65)
```

In [32]:

```
min(tup3)
```

Out[32]:

```
10
```

In [33]:

```
max(tup3)
```

Out[33]:

```
65
```

2--lists

- ordered collection of elements
- enclosed in [] square brackets / parenthesis
- mutateable / you can change the values

In [34]:

```
list1 = [1, "python", True, 5]
list1
```

Out[34]:

```
[1, 'python', True, 5]
```

In [35]:

```
type(list1)
```

Out[35]:

```
list
```

In [36]:

```
list1[2]
```

Out[36]:

```
True
```

In [37]:

```
list2 =[3, 5, "ammar", "codanics", 478, 53.2, False]
list2
```

Out[37]:

```
[3, 5, 'ammar', 'codanics', 478, 53.2, False]
```

In [40]:

```
list2.append("codanics youtube channel")
list2
```

Out[40]:

```
[3,
 5,
 'ammar',
 'codanics',
 478,
 53.2,
 False,
 'codanics youtube channel',
 'codanics youtube channel']
```

In [42]:

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

```
Out[42]: [1, 'python', True, 5, 'codanics youtube channel', 'codanics youtube channel']
```

```
In [43]: list1+list2
```

```
Out[43]: [1,
          'python',
          True,
          5,
          'codanics youtube channel',
          'codanics youtube channel',
          3,
          5,
          'ammar',
          'codanics',
          478,
          53.2,
          False,
          'codanics youtube channel',
          'codanics youtube channel']
```

```
In [44]: list2.reverse()
         list2
```

```
Out[44]: ['codanics youtube channel',
          'codanics youtube channel',
          False,
          53.2,
          478,
          'codanics',
          'ammar',
          5,
          3]
```

```
In [45]: list1.count("5:5")
         list1
```

```
Out[45]: [1, 'python', True, 5, 'codanics youtube channel', 'codanics youtube channel']
```

```
In [59]: list1.remove("python")
         list1
```

```
Out[59]: [1, True, 5, 'codanics youtube channel', 'codanics youtube channel']
```

```
In [61]: list1.append ("python")
         list1
```

```
Out[61]: [1, True, 5, 'codanics youtube channel', 'codanics youtube channel', 'python']
```

```
In [63]: list3=[10,15,12,20,18,25,22,17]
         list3
```

```
Out[63]: [10, 15, 12, 20, 18, 25, 22, 17]
```

```
In [68]: list3.count("0:10")
         list3

Out[68]: [10, 15, 12, 20, 18, 25, 22, 17]

In [69]: len(list3)

Out[69]: 8

In [70]: list3.sort()
         list3

Out[70]: [10, 12, 15, 17, 18, 20, 22, 25]

In [71]: list3*3

Out[71]: [10,
          12,
          15,
          17,
          18,
          20,
          22,
          25,
          10,
          12,
          15,
          17,
          18,
          20,
          22,
          25,
          10,
          12,
          15,
          17,
          18,
          20,
          22,
          25]

In [72]: list3*2

Out[72]: [10, 12, 15, 17, 18, 20, 22, 25, 10, 12, 15, 17, 18, 20, 22, 25]

In [73]: list1+list2+list3

Out[73]: [1,
          True,
          5,
          'codanics youtube channel',
          'codanics youtube channel',
          'python',
          'codanics youtube channel',
          'codanics youtube channel',
 False,
 53.2,
 478,
 'codanics',
 'ammar',
 5,
 3,
 10,
 12,
 15,
 17,
 18,
 20,
 22,
 25]
```

3 -Dictionaries

- an unordered collection of elements
- consist of two things. key & value
- we will use curly braces to make dictionaries
- Mutateable / you can change the values

In [75]:

```
#Food and their prices
food1 = {"samosa":30, "pakora":100, "raita":20, "salad":20, "chicken rolls":30}
food1
```

Out[75]:

```
{'samosa': 30, 'pakora': 100, 'raita': 20, 'salad': 20, 'chicken rolls': 30}
```

In [76]:

```
type(food1)
```

Out[76]:

```
dict
```

In [79]:

```
# extract data
keys1 = food1.keys()
keys1
```

Out[79]:

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

In [80]:

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

Out[80]:

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

In [90]:

```
# adding an element
food1["tikki"] = 10
food1
```

Out[90]:

```
{'samosa': 30,
 'pakora': 100,
 'raita': 20,
 'salad': 20,
 'chicken rolls': 30,
 'tikke': 10,
 'tikki': 10}
```

In [93]:

```
# update the values
food1["tikki"]=15
food1
```

Out[93]:

```
{'samosa': 30,
 'pakora': 100,
 'raita': 20,
 'salad': 20,
 'chicken rolls': 30,
 'tikke': 10,
 'tikki': 15}
```

In [97]:

```
food2 = {"dates":50, "swayan":100, "chocolates":80,}
food2
```

Out[97]:

```
{'dates': 50, 'swayan': 100, 'chocolates': 80}
```

In [98]:

```
type(food2)
```

Out[98]:

```
dict
```

In [100…

```
# how to concatinate dictionaries
food1.update(food2)
food1
```

Out[100…

```
{'samosa': 30,
 'pakora': 100,
 'raita': 20,
 'salad': 20,
 'chicken rolls': 30,
 'tikke': 10,
 'tikki': 15,
 'dates': 50,
 'swayan': 100,
 'chocolates': 80}
```

4-Sets

- unordered and unindexed collection of elements
- curly braces {} are used to store the elements
- no duplicates allowed

In [101…

```
s1 = {1,2,2,5,2, "Ammar", "codanics", "Faisalabad"}
s1
```

Out[101…

```
{1, 2, 5, 'Ammar', 'Faisalabad', 'codanics'}
```

In [106…

```
type(s1)
```

```
Out[106… set
```

In [107…

```
s1.add("Faisalabad")
s1
```

```
Out[107… {1, 2, 5, 'Ammar', 'Faisalabad', 'ammar', 'codanics'}
```

In [109…

```
s1.add("faisalabad")
s1
```

```
Out[109… {1, 2, 5, 'Ammar', 'Faisalabad', 'ammar', 'codanics', 'faisalabad'}
```

In [114…

```
s1.difference("ammar", "Ammar")
s1
```

```
Out[114… {2, 5, 'Ammar', 'Faisalabad', 'ammar', 'codanics', 'faisalabad'}
```

In [115…

```
s1.pop()
s1
```

```
Out[115… {2, 5, 'Faisalabad', 'ammar', 'codanics', 'faisalabad'}
```

In [117…

```
s1.isdisjoint("ammar")
s1
```

```
Out[117… {2, 5, 'Faisalabad', 'ammar', 'codanics', 'faisalabad'}
```

In [120…

```
s1.isdisjoint("faisalabad")
s1
```

```
Out[120… {2, 5, 'Faisalabad', 'ammar', 'codanics', 'faisalabad'}
```

In [ ]: