python ka chila day 3 task

- indexing
- sring method
- tuple and there function
- list and there fuctions
- dictionary and there functions
- set and there function

indexing

```
In [1]: x = "apple juice" ## write a string
```

```
In [2]: x
Out[2]: 'apple juice'
```

```
In [3]: x[3] #finding the character through character
Out[3]: 'l'
```

```
In [4]: len(x) ## finding the lenght of string`
Out[4]: 11
```

```
In [5]: x[12] ## see the error arrise becoz indexing start form x[0]

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-5-d499977f0e23> in <module>
----> 1 x[12] ## see the error arrise becoz indexing start form x[0]

IndexError: string index out of range
```

```
In [6]: x[5]
Out[6]: ' '
```

```
In [7]: x[0]
Out[7]: 'a'
```

In [8]:
```
##if take the one whole word in the string but full form not show bocoz las
t one exclude
x[0:4]
## false method of doing the code x[0to5]
```

Out[8]:
```
'appl'
```

In [9]:
```
x[0:5] ## correct method of show the word
```

Out[9]:
```
'apple'
```

In [10]:
```
## printing the value form the - side
x[-6:-1] ## half printing
```

Out[10]:
```
' juic'
```

In [11]:
```
x[-6:12] ## correct method
```

Out[11]:
```
' juice'
```

string methods

In [12]:
```
food = "biryani"
```

In [13]:
```
len(food)
```

Out[13]:
```
7
```

In [14]:
```
food.capitalize()
```

Out[14]:
```
'Biryani'
```

In [15]:
```
food.upper()
```

Out[15]:
```
'BIRYANI'
```

In [16]:
```
food.lower()
```

Out[16]:
```
'biryani'
```

In [17]:
```
food.replace("b","sh")## replace the letter to new one
```

Out[17]:
```
'shiryani'
```

In [18]:
```
## counting the specific letter
name = "baba ammar dr tufail"
```

In [19]:
```
name.count("a")
```

Out[19]:
```
5
```

finding index number

```
In [20]: name = "baba ammar dr tufail"
```

```
In [21]: name.find('m')
```

```
Out[21]: 6
```

```
In [22]: ## split the string
         khana = "roti a chawal a pani a pepsi "
         khana
```

```
Out[22]: 'roti a chawal a pani a pepsi '
```

```
In [23]: khana.split("a" ) ## it is spliting on the a
```

```
Out[23]: ['roti ', ' ch', 'w', 'l ', ' p', 'ni ', ' pepsi ']
```

---

tuple

- orderd collection
- enclose()
- different element store
- unmutable

```
In [24]: tup = (1,3,"adc")
         tp =(9,"dd",2)
         tp+tup
```

```
Out[24]: (9, 'dd', 2, 1, 3, 'adc')
```

indexing

```
In [25]: tp[1]
```

```
Out[25]: 'dd'
```

```
In [26]: tp[0:3]
```

```
Out[26]: (9, 'dd', 2)
```

```
In [27]: len(tp)

Out[27]: 3
```

```
In [28]: type(tp)

Out[28]: tuple
```

multiply the tuple add them

```
In [29]: tp*2+tup

Out[29]: (9, 'dd', 2, 9, 'dd', 2, 1, 3, 'adc')
```

```
In [30]: tp.count("dd")

Out[30]: 1
```

```
In [31]: (tp+tup)*3

Out[31]: (9, 'dd', 2, 1, 3, 'adc', 9, 'dd', 2, 1, 3, 'adc', 9, 'dd', 2, 1, 3, 'adc')
```

```
In [32]: tp2=(12,3,4,224,5,5,2,3)
```

```
In [33]: min(tp2)## show the min values

Out[33]: 2
```

```
In [34]: max(tp2)## higest values

Out[34]: 224
```

list

- orded collection
- enclosed []
- mutatable

```
In [35]: list=[1,2,"apple"]
         list

Out[35]: [1, 2, 'apple']
```

```
In [36]: len(list)

Out[36]: 3
```

```
In [37]:  list[1]## accesing by the index

Out[37]: 2
```

functions in list

```
In [38]:  list.append('ok') ## adding something in list
          list

Out[38]: [1, 2, 'apple', 'ok']

In [39]:  list.count('ok') ## checking the thing how many time append

Out[39]: 1

In [40]:  list

Out[40]: [1, 2, 'apple', 'ok']

In [41]:  x=list.copy() ## copy the list into new list

In [42]:  x

Out[42]: [1, 2, 'apple', 'ok']

In [43]:  list.extend('apple') ## itrate the string in the characters

In [44]:  list

Out[44]: [1, 2, 'apple', 'ok', 'a', 'p', 'p', 'l', 'e']

In [45]:  list.insert(3,"new thing")## insert the new in specific index

In [46]:  list

Out[46]: [1, 2, 'apple', 'new thing', 'ok', 'a', 'p', 'p', 'l', 'e']

In [47]:  list.pop() ## remove the last element in the list and print on tab

Out[47]: 'e'

In [48]:  list.remove("new thing") ## put the name of which u remove

In [49]:  list

Out[49]: [1, 2, 'apple', 'ok', 'a', 'p', 'p', 'l']
```

In [50]:
```
list.reverse() reverse the order
```

```
  File "<ipython-input-50-8b757529e1c2>", line 1
    list.reverse() reverse the order
                         ^
SyntaxError: invalid syntax
```

In [51]:
```
list
```

Out[51]:
```
[1, 2, 'apple', 'ok', 'a', 'p', 'p', 'l']
```

In [52]:
```
list.sort() ## only work for the homogeniuos data not hetro
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-52-b379d69e64cd> in <module>
----> 1 list.sort() ## only work for the homogeniuos data not hetro

TypeError: '<' not supported between instances of 'str' and 'int'
```

In [53]:
```
l = [2,4,4,532,3,5,2]
```

In [54]:
```
l
```

Out[54]:
```
[2, 4, 4, 532, 3, 5, 2]
```

In [55]:
```
l.sort()## now sort in the orderr
```

In [56]:
```
l
```

Out[56]:
```
[2, 2, 3, 4, 4, 5, 532]
```

In [57]:
```
list.clear()  ##clear every thing
list
```

Out[57]:
```
[]
```

dictonary

- key value pair
- {}
- mutatable

In [58]:
```
dic1 ={"kela":40,"saib":60,12:90}
dic1
```

Out[58]:
```
{'kela': 40, 'saib': 60, 12: 90}
```

fuctions

In [59]:
```
#keys
dic1.keys()
```

Out[59]:
```
dict_keys(['kela', 'saib', 12])
```

In [60]:
```
dic1.values() ## provide values
```

Out[60]:
```
dict_values([40, 60, 90])
```

In [61]:
```
## return values given key
dic1.get(12)
```

Out[61]:
```
90
```

In [62]:
```
dic1.popitem() ## delet last item
```

Out[62]:
```
(12, 90)
```

In [63]:
```
dic1.pop("kela") ## by giving the key delete both key value pairs
```

Out[63]:
```
40
```

In [64]:
```
a=dic1.fromkeys('kela',1) ## break into new dictionaery
a
```

Out[64]:
```
{'k': 1, 'e': 1, 'l': 1, 'a': 1}
```

In [65]:
```
dic1.update({67:99})## puting new value
```

In [66]:
```
dic1.items() ## both pair
```

Out[66]:
```
dict_items([('saib', 60), (67, 99)])
```

In [67]:
```
dic1.setdefault(112)## set the default value is none u not give it
dic1
```

Out[67]:
```
{'saib': 60, 67: 99, 112: None}
```

In [68]:
```
newdic=dic1.copy() ## copy to new
```

In [69]:
```
newdic
```

Out[69]:
```
{'saib': 60, 67: 99, 112: None}
```

concatinate 2 dictionaries

```
In [70]: a={"ab":2,"cd":5,"fr":9}
         b={"apple":49,"charry":70,"orange":90}
```

```
In [71]: a.update(b)
         a
```

```
Out[71]: {'ab': 2, 'cd': 5, 'fr': 9, 'apple': 49, 'charry': 70, 'orange': 90}
```

---

sets

- unorder un index
- use in {}
- no duplicates
- no key value

```
In [72]: s={1,889,"ali","kpk","Flase" }
```

set functions

```
In [73]: s.update("fatmi") ## add string in character form
         s
```

```
Out[73]: {1, 889, 'Flase', 'a', 'ali', 'f', 'i', 'kpk', 'm', 't'}
```

```
In [74]: s.add("mango")
```

```
In [75]: s
```

```
Out[75]: {1, 889, 'Flase', 'a', 'ali', 'f', 'i', 'kpk', 'm', 'mango', 't'}
```

```
In [76]: s.discard('i') ## romve the element in the set
         s
```

```
Out[76]: {1, 889, 'Flase', 'a', 'ali', 'f', 'kpk', 'm', 'mango', 't'}
```

```
In [77]: s.pop() ## remove the element without giving the arugments
```

```
Out[77]: 1
```

In [78]:
```
s.remove("f") ## remove the element in the set
s
```

Out[78]:
```
{889, 'Flase', 'a', 'ali', 'kpk', 'm', 'mango', 't'}
```

In [79]:
```
a = {1,2,3,4,5}
b= {4,5,6,7,}
```

In [80]:
```
a.difference(b) ## which things are not in b
```

Out[80]:
```
{1, 2, 3}
```

In [81]:
```
a.union(b)
```

Out[81]:
```
{1, 2, 3, 4, 5, 6, 7}
```

In [82]:
```
b.intersection(a)
```

Out[82]:
```
{4, 5}
```

In [83]:
```
a
```

Out[83]:
```
{1, 2, 3, 4, 5}
```

In [84]:
```
a.discard(1) ## delete the values
```

In [85]:
```
a
```

Out[85]:
```
{2, 3, 4, 5}
```

In [86]:
```
x ={1,3}
y = {2,4}
x.isdisjoint(y) ## if there is no common values in the set provide true else false
```

Out[86]:
```
True
```

In [87]:
```
a.issubset(b) ## show the sub-set of set
```

Out[87]:
```
False
```

In [88]:
```
a.intersection_update(b)## find intersection and delete values which are not common in "a"
a
```

Out[88]:
```
{4, 5}
```

In [89]:

```
 b.difference_update(a) ## check differnce and delete similar values which are in other set
print(b)
print(a)
```

```
{6, 7}
{4, 5}
```

task finish of day 3