INDEXING

Making a string

In [1]:
```
a ="custurd triffle"
```

In [2]:
```
a
```

Out[2]:
```
'custurd triffle'
```

Length of the string

In [3]:
```
len(a)
```

Out[3]:
```
15
```

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

Out[4]:
```
'c'
```

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

Out[5]:
```
'u'
```

In [6]:
```
a[2]
```

Out[6]:
```
's'
```

In [7]:
```
a[14]
```

Out[7]:
```
'e'
```

last indeces are exclusive

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

Out[8]:
```
'custurd '
```

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

Out[9]:
```
'e'
```

In [10]:
```
a[-7]
```

Out[10]:
```
't'
```

In [11]:

```
a[-7:15]
```

Out[11]: 'triffle'

STRINGS METHODS

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

In [13]:
```
food
```

Out[13]: 'raita'

In [14]:
```
len(food)
```

Out[14]: 5

In [15]:
```
#capitalize each word
food.capitalize()
```

Out[15]: 'Raita'

In [16]:
```
#uppercase whole
food.upper()
```

Out[16]: 'RAITA'

In [17]:
```
#lowercase whole
food.lower()
```

Out[17]: 'raita'

In [18]:
```
#relace
food.replace("r","sh")
```

Out[18]: 'shaita'

In [19]:
```
myself="I am a Phd student of bioinformatics from China currently suspended due to C
```

In [20]:
```
myself
```

Out[20]: 'I am a Phd student of bioinformatics from China currently suspended due to Covid'

In [21]:
```
#counting a specific letter
myself.count("a")
```

Out[21]: 4

In [22]:

```
myself.count("C")
```

Out[22]: 2

Finding an index number in a string

In [23]:
```
myself="I am a Phd student of bioinformatics from China currently suspended due to C
```

In [24]:
```
myself
```

Out[24]: 'I am a Phd student of bioinformatics from China currently suspended due to Covid'

In [25]:
```
myself.find("bio")
```

Out[25]: 22

In [26]:
```
myself.find("Covid")
```

Out[26]: 75

how to split a string

In [27]:
```
myself.split(" ")
```

Out[27]:
```
['I',
 'am',
 'a',
 'Phd',
 'student',
 'of',
 'bioinformatics',
 'from',
 'China',
 'currently',
 'suspended',
 'due',
 'to',
 'Covid']
```

Basics DATA STRUCTURES in Python

1. tuples
2. Lists
3. Dictionaries
4. Sets

1-TUPLE

1. ordered collection of elements
2. enclosed in ()

3. different elements can be stored
4. unmutatable

In [28]:
```
tup1=(43,"python ka chilla",False,8.7)
tup1
```

Out[28]:
```
(43, 'python ka chilla', False, 8.7)
```

indexing in tuple

In [29]:
```
tup1[1]
```

Out[29]:
```
'python ka chilla'
```

In [30]:
```
tup1[0:5]
```

Out[30]:
```
(43, 'python ka chilla', False, 8.7)
```

In [31]:
```
tup2=(8.5,True,"azka junaid",56)
tup2
```

Out[31]:
```
(8.5, True, 'azka junaid', 56)
```

In [32]:
```
#concatinate tuple
tup1+tup2
```

Out[32]:
```
(43, 'python ka chilla', False, 8.7, 8.5, True, 'azka junaid', 56)
```

In [33]:
```
#concatinate and repeat
tup1*2+tup2
```

Out[33]:
```
(43,
 'python ka chilla',
 False,
 8.7,
 43,
 'python ka chilla',
 False,
 8.7,
 8.5,
 True,
 'azka junaid',
 56)
```

In [34]:
```
len(tup1*2+tup2)
```

Out[34]:
```
12
```

In [35]:
```
tup3=(34,56,82,19)
tup3
```

Out[35]:
```
(34, 56, 82, 19)
```

In [36]:
```
#minimun in tuple
min(tup3)
```

Out[36]:
```
19
```

In [37]:
```
#maximum in tuple
max(tup3)
```

Out[37]:
```
82
```

In [86]:
```
tup3
```

Out[86]:
```
(34, 56, 82, 19)
```

2- LISTS

1. ordered collection of different elements
2. enclosed in []
3. mutatable/changeable

In [39]:
```
list1=[8.5,True,"azka junaid",56]
list1
```

Out[39]:
```
[8.5, True, 'azka junaid', 56]
```

In [40]:
```
list1[3]
```

Out[40]:
```
56
```

In [41]:
```
len(list1)
```

Out[41]:
```
4
```

In [42]:
```
list1[3]
```

Out[42]:
```
56
```

In [43]:
```
list2=["azka","aammar",88,77.8,77,9865,False]
list2
```

Out[43]:
```
['azka', 'aammar', 88, 77.8, 77, 9865, False]
```

In [44]:
```
list1+list2
```

Out[44]:
```
[8.5, True, 'azka junaid', 56, 'azka', 'aammar', 88, 77.8, 77, 9865, False]
```

In [45]:

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

Out[45]:
```
[8.5, True, 'azka junaid', 56, 8.5, True, 'azka junaid', 56]
```

In [46]:
```
list3=list1+list2
list3
```

Out[46]:
```
[8.5, True, 'azka junaid', 56, 'azka', 'aammar', 88, 77.8, 77, 9865, False]
```

In [47]:
```
list3.append("the")
list3
```

Out[47]:
```
[8.5,
 True,
 'azka junaid',
 56,
 'azka',
 'aammar',
 88,
 77.8,
 77,
 9865,
 False,
 'the']
```

In [48]:
```
list3.clear()
```

In [49]:
```
list2.copy()
```

Out[49]:
```
['azka', 'aammar', 88, 77.8, 77, 9865, False]
```

In [50]:
```
list2
```

Out[50]:
```
['azka', 'aammar', 88, 77.8, 77, 9865, False]
```

In [51]:
```
list2.reverse()
list2
```

Out[51]:
```
[False, 9865, 77, 77.8, 88, 'aammar', 'azka']
```

In [52]:
```
list3=[34,87,85,6,6,6,3,48]
list3.sort()
list3
```

Out[52]:
```
[3, 6, 6, 6, 34, 48, 85, 87]
```

assignment query

In [53]:
```
list3.count(6)
```

Out[53]:
```
3
```

In [54]:

```
list2.count("aammar")
```

Out[54]: 1

3-DICTIONSRIES

1. unordered collection of elements
2. keys and values
3. round braces {}
4. mutatable**

In [55]:
```
makeup={"mascara":700,"lipstick":600,"lipmatte":800,"foundation":1800,"blush":500}
makeup
```

Out[55]:
```
{'mascara': 700,
 'lipstick': 600,
 'lipmatte': 800,
 'foundation': 1800,
 'blush': 500}
```

In [56]:
```
type(makeup)
```

Out[56]: dict

In [57]:
```
keysmakeup=makeup.keys()
keysmakeup
```

Out[57]: dict_keys(['mascara', 'lipstick', 'lipmatte', 'foundation', 'blush'])

In [58]:
```
valuemakeup=makeup.values()
valuemakeup
```

Out[58]: dict_values([700, 600, 800, 1800, 500])

In [59]:
```
makeup["liner"]=450
```

In [60]:
```
makeup.update
```

Out[60]: <function dict.update>

In [63]:
```
makeup
```

Out[63]:
```
{'mascara': 700,
 'lipstick': 600,
 'lipmatte': 800,
 'foundation': 1800,
 'blush': 500,
 'liner': 450}
```

In [61]:
```
makeover={"dress":4500,"chappal":1700,"jewelery":700,"bag":3000}
makeover
```

Out[61]:
```
{'dress': 4500, 'chappal': 1700, 'jewelery': 700, 'bag': 3000}
```

In [62]:
```
makeover.update(makeup)
makeover
```

Out[62]:
```
{'dress': 4500,
 'chappal': 1700,
 'jewelery': 700,
 'bag': 3000,
 'mascara': 700,
 'lipstick': 600,
 'lipmatte': 800,
 'foundation': 1800,
 'blush': 500,
 'liner': 450}
```

In [64]:
```
makeover.values()
```

Out[64]:
```
dict_values([4500, 1700, 700, 3000, 700, 600, 800, 1800, 500, 450])
```

In [65]:
```
makeovervalue= sum(makeover.values())
makeovervalue
```

Out[65]:
```
14750
```

4-SETS

1. unordered unindexed collection
2. curly braces{}
3. no duplicate**

In [66]:
```
set1={66,87.9,76,34,54,"joke funny tha"}
set1
```

Out[66]:
```
{34, 54, 66, 76, 87.9, 'joke funny tha'}
```

In [67]:
```
set1.add(61)
set1
```

Out[67]:
```
{34, 54, 61, 66, 76, 87.9, 'joke funny tha'}
```

In [68]:
```
set1.clear()
set1
```

Out[68]:
```
set()
```

In [69]:
```
set1={66,87.9,76,34,54,"joke funny tha"}
set1
```

Out[69]:
```
{34, 54, 66, 76, 87.9, 'joke funny tha'}
```

In [70]:

```
set1.copy()
```

Out[70]: {34, 54, 66, 76, 87.9, 'joke funny tha'}

In [71]:
```
set2={67,84,66,54,92}
set2
```

Out[71]: {54, 66, 67, 84, 92}

In [72]:
```
set1.difference(set2)
```

Out[72]: {34, 76, 87.9, 'joke funny tha'}

In [73]:
```
set1.difference_update(set2)
set1
```

Out[73]: {34, 76, 87.9, 'joke funny tha'}

In [74]:
```
set1.discard(76)
set1
```

Out[74]: {34, 87.9, 'joke funny tha'}

In [85]:
```
set1.intersection()
```

Out[85]: {34, 87.9, 'joke funny tha'}