```
- Data Structures
    - tuple
    - list
    - dictionary
    - set
```

```
### list()
    - It is used to store multiple items in a single variable
    - It can allow the duplicates
    - It is mutable
    (mutable:-  we can change the data)
    - We can store any type of data
    - Represented by [], values can separated by ,

```

In [2]:

```
li = []
print(li,type(li))
```

```
[] <class 'list'>
```

In [3]:

```
li1 = [7,5,9,"a","v",7.9,3.7]
li1
```

Out[3]:

```
[7, 5, 9, 'a', 'v', 7.9, 3.7]
```

In [4]:

```
li1[0]
```

Out[4]:

```
7
```

In [5]:

```
li1[-1]
```

Out[5]:

```
3.7
```

In [7]:

```
li1[1:6]
```

Out[7]:

```
[5, 9, 'a', 'v', 7.9]
```

In [9]:

```
li1[::-1]
```

Out[9]:

```
[3.7, 7.9, 'v', 'a', 9, 5, 7]
```

In [12]:

```
# min(),max(),sum(),len(),sorted()
li2 = [34,11,67,7.8,1.7,88]
print(min(li2),max(li2),sum(li2))
print(sorted(li2),len(li2))
```

```
1.7 88 209.5
[1.7, 7.8, 11, 34, 67, 88] 6
```

In [13]:

```
k = [6,8,5,[85,78,33],89]
k
```

Out[13]:

```
[6, 8, 5, [85, 78, 33], 89]
```

In [14]:

```
k[3]
```

Out[14]:

```
[85, 78, 33]
```

In [15]:

```
k[3][2]
```

Out[15]:

```
33
```

In [16]:

```
print(dir(list),end = ' ')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__
dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__',
'__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reverse
d__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index',
'insert', 'pop', 'remove', 'reverse', 'sort']
```

In [17]:

```
# append()
li1 = [8,9,34,55]
li1.append(56)
li1
```

Out[17]:

```
[8, 9, 34, 55, 56]
```

In [18]:

```
# count()
print(li1.count(55))
```

```
1
```

In [19]:

```
# index()
print(li1.index(9))
```

```
1
```

In [20]:

```
# extend()
a = [2,3,5,6]
b = [3,5,8,0]
a.extend(b)
print(a)
```

```
[2, 3, 5, 6, 3, 5, 8, 0]
```

In [21]:

```
# insert()
a.insert(2,23)
print(a)
```

```
[2, 3, 23, 5, 6, 3, 5, 8, 0]
```

In [22]:

```
# pop()
print(a.pop())
```

```
0
```

In [23]:

```
print(a)
```

```
[2, 3, 23, 5, 6, 3, 5, 8]
```

In [24]:

```
a.remove(6)
```

In [25]:

```
a
```

Out[25]:

```
[2, 3, 23, 5, 3, 5, 8]
```

In [30]:

```
# copy()
s1 = [11,34,23,67]
s2 = s1.copy()
print(s2,s1)
s2.pop()
print(s2,s1)
```

```
[11, 34, 23, 67] [11, 34, 23, 67]
[11, 34, 23] [11, 34, 23, 67]
```

In [29]:

```
x = [1,2,3,4]
y = x
print(x,y)
y.pop()
print(y,x)
```

```
[1, 2, 3, 4] [1, 2, 3, 4]
[1, 2, 3] [1, 2, 3]
```

In [31]:

```
a
```

Out[31]:

```
[2, 3, 23, 5, 3, 5, 8]
```

In [32]:

```
a.clear()
```

In [33]:

```
a
```

Out[33]:

```
[]
```

In [34]:

```
s = [2,3,5,6]
for i in s:
    print(i,end=' ')
```

```
2 3 5 6
```

In [35]:

```
s = [2,3,5,6]
for i in s:
    if(i%2!=0):
        print(i,end=' ')
```

```
3 5
```

In [36]:

```
t = 0
for i in s:
    t = t+i
print(t)
```

```
16
```

In [ ]:

```
# i/p:  [243,56,11,23,11,23,67,19]
# o/p:  [243,56,11,23,67,19]
```

In [40]:

```
li1 = [243,56,11,23,11,23,67,19]
li2 = []
for i in li1:  # 243 56 11 23 11 23 67 19
    if i not in li2:  # 243
        li2.append(i) # 243  56 11 23 67 19
print(li2)
```

```
[243, 56, 11, 23, 67, 19]
```

```
#### dictionary
    - It is a key-value pair
    - It is mutable
    - It is a collection which is ordered and changeable
    - Keys does not allow the duplicates
    - values can allow the duplicates
    - Represented by curly brackets {}
```

In [41]:

```
d = {}
print(d,type(d))
```

```
{} <class 'dict'>
```

In [42]:

```
d = {2:45,6:89,3:9,"a":66,"b":"c"}
d
```

Out[42]:

```
{2: 45, 6: 89, 3: 9, 'a': 66, 'b': 'c'}
```

In [46]:

```
print(d[2])
print(d["a"])
```

```
45
66
```

In [48]:

```
d = {2:45,6:89,3:9,"a":66,2:90,"b":"c",2:67,7:89}
d
```

Out[48]:

```
{2: 67, 6: 89, 3: 9, 'a': 66, 'b': 'c', 7: 89}
```

In [49]:

```
print(dir(dict),end=' ')
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__
doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem_
_', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__
le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_e
x__', '__repr__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys',
'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [51]:

```
# get
print(d)
d.get(7)
```

```
{2: 67, 6: 89, 3: 9, 'a': 66, 'b': 'c', 7: 89}
```

Out[51]:

```
89
```

In [54]:

```
# keys(),values(),items()
print(d.keys())
print(d.values())
print(d.items())
```

```
dict_keys([2, 6, 3, 'a', 'b', 7])
dict_values([67, 89, 9, 66, 'c', 89])
dict_items([(2, 67), (6, 89), (3, 9), ('a', 66), ('b', 'c'), (7, 89)])
```

In [55]:

```
d
```

Out[55]:

```
{2: 67, 6: 89, 3: 9, 'a': 66, 'b': 'c', 7: 89}
```

In [56]:

```
d["da"]="workshop"
d
```

Out[56]:

```
{2: 67, 6: 89, 3: 9, 'a': 66, 'b': 'c', 7: 89, 'da': 'workshop'}
```

In [57]:

```
# pop()
d.pop('a')
```

Out[57]:

```
66
```

In [58]:

```
d
```

Out[58]:

```
{2: 67, 6: 89, 3: 9, 'b': 'c', 7: 89, 'da': 'workshop'}
```

In [59]:

```
# popitem()
d.popitem()
```

Out[59]:

```
('da', 'workshop')
```

In [60]:

```
print(d)
```

```
{2: 67, 6: 89, 3: 9, 'b': 'c', 7: 89}
```

In [63]:

```
# update()
d.update({'b':"ab"})
```

In [64]:

```
d
```

Out[64]:

```
{2: 67, 6: 89, 3: 9, 'b': 'ab', 7: 89}
```

In [65]:

```
d.clear()
```

In [66]:

```
d
```

Out[66]:

```
{}
```

In [67]:

```
li = [1,2,3,4,5]
d = {}
for i in li:
    d[i]=i**2
print(d)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

In [ ]:

```
# 10
# {1:1, 2:8, 3:27..............10:1000}
```

```
## set()
- It does not allow the duplicates
- Set itself mutable
- Represented by {}
```

In [69]:

```
s = {3,4}
print(s,type(s))
```

```
{3, 4} <class 'set'>
```

In [70]:

```
s = {3,4,5,1,2,3,4,5,9}
s
```

Out[70]:

```
{1, 2, 3, 4, 5, 9}
```

In [71]:

```
print(min(s),max(s),sum(s),sorted(s),len(s))
```

```
1 9 24 [1, 2, 3, 4, 5, 9] 6
```

In [72]:

```
print(dir(set),end=' ')
```

```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__has
h__', '__iand__', '__init__', '__init_subclass__', '__ior__', '__isub__',
'__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new_
_', '__or__', '__rand__', '__reduce__', '__reduce_ex__', '__repr__', '__ro
r__', '__rsub__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__s
ub__', '__subclasshook__', '__xor__', 'add', 'clear', 'copy', 'differenc
e', 'difference_update', 'discard', 'intersection', 'intersection_update',
'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric_differ
ence', 'symmetric_difference_update', 'union', 'update']
```

In [73]:

```
s1 = {3,34,12,67}
s1.add(45)
s1
```

Out[73]:

```
{3, 12, 34, 45, 67}
```

In [74]:

```
# union()
a,b={1,2,45,66},{34,11,56,2,1}
a.union(b)
```

Out[74]:

```
{1, 2, 11, 34, 45, 56, 66}
```

In [75]:

```
# intersection()
a.intersection(b)
```

Out[75]:

```
{1, 2}
```

In [76]:

```
b.intersection(a)
```

Out[76]:

```
{1, 2}
```

In [77]:

```
# difference()
a.difference(b)
```

Out[77]:

```
{45, 66}
```

In [78]:

```
b.difference(a)
```

Out[78]:

```
{11, 34, 56}
```

In [79]:

```
# discard()
s = {56,12,66,34}
s.discard(12)
```

In [80]:

```
s
```

Out[80]:

```
{34, 56, 66}
```

In [81]:

```
s.remove(12)
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
<ipython-input-81-f2b88b478e9d> in <module>
----> 1 s.remove(12)

KeyError: 12
```

In [82]:

```
s.remove(66)
```

In [83]:

```
s
```

Out[83]:

```
{34, 56}
```

In [84]:

```
s.discard(66)
```

In [85]:

```
s
```

Out[85]:

```
{34, 56}
```

In [87]:

```
x = {"a","b","c"}
y = {"d","e","a"}
z = x.isdisjoint(y)
print(z)
```

```
False
```

In [ ]:

```
# s = {4,2,6,11,78,9}
# {16,4,36}
```

In [88]:

```
# i/p :- [1,23,2,3,56,56,78,1,23]
# o/p:- [2,3,78]
li = [1,23,2,3,56,56,78,1,23]
li1 = []
for i in li:
    if(li.count(i)==1):
        li1.append(i)
print(li1)
```

```
[2, 3, 78]
```

In [ ]:

```

```

In [ ]:

```

```