Datastaracture-LIST-TUPLE

```
In [7]: l=[10,20,30,40,50]
        print(l)
```

```
[10, 20, 30, 40, 50]
```

```
In [8]: l.reverse()
```

```
In [9]: l
```

```
Out[9]: [50, 40, 30, 20, 10]
```

```
In [10]: l.sort() #Ascending order means smaller to larger
         print(l)
```

```
[10, 20, 30, 40, 50]
```

```
In [11]: l.sort(reverse=True) #Descending means larger to smaller
         print(l)
```

```
[50, 40, 30, 20, 10]
```

```
In [14]: import keyword
         keyword.kwlist
```

```
Out[14]: ['False',
          'None',
          'True',
          'and',
          'as',
          'assert',
          'async',
          'await',
          'break',
          'class',
          'continue',
          'def',
          'del',
          'elif',
          'else',
          'except',
          'finally',
          'for',
          'from',
          'global',
          'if',
          'import',
          'in',
          'is',
          'lambda',
          'nonlocal',
          'not',
          'or',
          'pass',
          'raise',
          'return',
          'try',
          'while',
          'with',
          'yield']
```

```
In [15]: l2=['2',2.3,1+2j,True]
         print(l2)
```

```
['2', 2.3, (1+2j), True]
```

```
In [17]: l2.sort() #sort function when you pass similar datatype
         print(l2)#parameter tuning-system given bydefult parameter
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 l2.sort()
      2 print(l2)

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

```
In [20]: l2.sort(reverse=True)#user change the system parameter
         print(l2)#hyperparameter tunning
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 l2.sort(reverse=True)
      2 print(l2)

TypeError: '<' not supported between instances of 'complex' and 'bool'
```

In [22]:
```
for i in l:
    print(i)
```

```
50
40
30
20
10
```

In [23]:
```
for i in enumerate(l):#it is take a 4 step before print(i)
    print(i)
```

```
(0, 50)
(1, 40)
(2, 30)
(3, 20)
(4, 10)
```

In [25]:
```
l1=[1,2,3,4,5]
print(l1)
```

```
[1, 2, 3, 4, 5]
```

In [26]:
```
for i in l1:print(i)
```

```
1
2
3
4
5
```

In [27]:
```
for i in enumerate(l1):print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
```

In [28]:
```
print(l)
print(l1)
```

```
[50, 40, 30, 20, 10]
[1, 2, 3, 4, 5]
```

In [29]:
```
all(l)
```

Out[29]:
```
True
```

In [30]:
```
any(l)
```

```
Out[30]: True
```

```
In [31]: l.sort()
         print(l)
```

```
[10, 20, 30, 40, 50]
```

```
In [32]: l.append(0)
         print(l)
```

```
[10, 20, 30, 40, 50, 0]
```

```
In [33]: all(l)
```

```
Out[33]: False
```

```
In [34]: any(l)
```

```
Out[34]: True
```

```
In [35]: l1
```

```
Out[35]: [1, 2, 3, 4, 5]
```

```
In [36]: l1[:]
```

```
Out[36]: [1, 2, 3, 4, 5]
```

```
In [37]: l1[3:]
```

```
Out[37]: [4, 5]
```

```
In [38]: l1[:3]
```

```
Out[38]: [1, 2, 3]
```

```
In [39]: l1=[1,10,15,20,25,30,40,50]
         print(l1)
```

```
[1, 10, 15, 20, 25, 30, 40, 50]
```

```
In [40]: l1.sort(reverse=True)
         print(l1)
```

```
[50, 40, 30, 25, 20, 15, 10, 1]
```

```
In [41]: l1[3:20]
```

```
Out[41]: [25, 20, 15, 10, 1]
```

```
In [42]: l1[20]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[42], line 1
----> 1 l1[20]

IndexError: list index out of range
```

In [43]:
```
l1
```

Out[43]:
```
[50, 40, 30, 25, 20, 15, 10, 1]
```

In [44]:
```
len(l1)
```

Out[44]:
```
8
```

In [45]:
```
l1[:5]
```

Out[45]:
```
[50, 40, 30, 25, 20]
```

In [46]:
```
l1[2:6]
```

Out[46]:
```
[30, 25, 20, 15]
```

In [47]:
```
l1[1:8:3]
```

Out[47]:
```
[40, 20, 1]
```

In [48]:
```
l1
```

Out[48]:
```
[50, 40, 30, 25, 20, 15, 10, 1]
```

In [49]:
```
l1[::2]
```

Out[49]:
```
[50, 30, 20, 10]
```

In [50]:
```
l1[::3]
```

Out[50]:
```
[50, 25, 10]
```

In [51]:
```
l1[::-1]# reverse printing
```

Out[51]:
```
[1, 10, 15, 20, 25, 30, 40, 50]
```

In [52]:
```
l1[::-2]
```

Out[52]:
```
[1, 15, 25, 40]
```

In [53]:
```
l1
```

Out[53]:
```
[50, 40, 30, 25, 20, 15, 10, 1]
```

In [54]:
```
l1[1:8:3]
```

```
Out[54]: [40, 20, 1]

In [55]: l1[5:-1]

Out[55]: [15, 10]

In [62]: l1

Out[62]: [50, 40, 30, 25, 20, 15, 10, 1]

In [64]: l1[0]=100
         print(l1)

[100, 40, 30, 25, 20, 15, 10, 1]

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

Tuple

```
In [ ]:

In [56]: t=()

In [57]: t

Out[57]: ()

In [58]: type(t)

Out[58]: tuple

In [59]: t=(10,20,30,40,50)
         print(t)

(10, 20, 30, 40, 50)

In [60]: t.index(40)
```

```
Out[60]:  3
```

```
In [61]:  t.count(10)
```

```
Out[61]:  1
```

```
In [66]:  t[0]=100
          print(t) # it is immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[66], line 1
----> 1 t[0]=100
      2 print(t)

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

```
In [67]:  t.append(10)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[67], line 1
----> 1 t.append(10)

AttributeError: 'tuple' object has no attribute 'append'
```

```
In [68]:  t.remove()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[68], line 1
----> 1 t.remove()

AttributeError: 'tuple' object has no attribute 'remove'
```

```
In [69]:  t1=(1,2,3,4)
          print(t1)
```

```
(1, 2, 3, 4)
```

```
In [71]:  t1[:2] # always silicing allowed here...
```

```
Out[71]:  (1, 2)
```

```
In [73]:  print(t)
          print(t1)
```

```
(10, 20, 30, 40, 50)
(1, 2, 3, 4)
```

```
In [77]:  a=sorted(t)
          print(a)
```

```
[10, 20, 30, 40, 50]
```

```
In [79]: a=sorted(t,reverse=True)
         print(a)

[50, 40, 30, 20, 10]

In [4]: t2=(1,2,3,4,5,6,7,8,9)
        print(t2)

(1, 2, 3, 4, 5, 6, 7, 8, 9)

In [5]: len(t2)

Out[5]: 9

In [7]: t2[::1]
        print(t2)

(1, 2, 3, 4, 5, 6, 7, 8, 9)

In [8]: t2[::-1]
        print(t2)

(1, 2, 3, 4, 5, 6, 7, 8, 9)

In [15]: t2[3:-7:-2]

Out[15]: (4,)

In [18]: t3=('i am javascript')
         print(t3)

i am javascript

In [19]: len(t3)

Out[19]: 15

In [20]: t3[1:8:2]

Out[20]: ' mjv'

In [21]: t3[3:7:5]

Out[21]: 'm'

In [24]: t3[4:9:3]

Out[24]: ' v'

In [30]: t3[7:-10:-3]

Out[30]: 'v'

In [33]: t3[6:-3]
```

```
Out[33]: 'avascr'

In [34]: t3[::3] # print evey 3rd index value

Out[34]: 'imasi'

In [36]: t3[::2]# print every 2nd index value

Out[36]: 'ia aacit'

In [37]: t3

Out[37]: 'i am javascript'
```

Tuple Creation

```
In [1]: t1=()
        print(t1)

()

In [2]: type(t1)

Out[2]: tuple

In [3]: (type(t1))

Out[3]: tuple

In [4]: t2=(10,20,30) #tuple of intigers numbers
        print(t2)

(10, 20, 30)

In [5]: t3=(10.2,20.3,30.4,40.5,50.5)
        print(t3) #tuple of float numbers

(10.2, 20.3, 30.4, 40.5, 50.5)

In [6]: t4=('one','two','three','four') #tuple of strings
        print(t4)

('one', 'two', 'three', 'four')

In [8]: t5=('asif',23,(10,20,30,40),(1+2j,2+3j)) # Nested tuples
        print(t5)

('asif', 23, (10, 20, 30, 40), ((1+2j), (2+3j)), True, False, 10.4)

In [9]: t6=('abhi',10,[50,100],[20,400],{'Ram','Lakhhman'},{99,22,33}) # tu[le of mixed dat
        print(t6)

('abhi', 10, [50, 100], [20, 400], {'Ram', 'Lakhhman'}, {33, 99, 22})
```

Tuple indexing

In [14]:
```
print (t2[0])# Retrive 1st 2nd 3rd element of the tuple
print (t2[1])
print (t2[2])
```
```
10
20
30
```

In [17]:
```
print (t3[0])
print (t3[1])
print (t3[2])
print (t3[3])
print (t3[4])
```
```
10.2
20.3
30.4
40.5
50.5
```

In [19]:
```
print (t4[0])
print (t4[1])
print (t4[2])
print (t4[3])
```
```
one
two
three
four
```

In [20]:
```
t2[-1]
```
Out[20]:
```
30
```

In [21]:
```
t2[-2]
```
Out[21]:
```
20
```

Tuple Slicing

In [22]:
```
mytuple=('one','two','three','four','five','six','seven','eight','nine')
print(mytuple)
```
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
```

In [23]:
```
mytuple[0:3]
```
Out[23]:
```
('one', 'two', 'three')
```

In [26]:
```
mytuple[5:1]
```

```
Out[26]: ()
```

```
In [27]: mytuple[2:5]
```

```
Out[27]: ('three', 'four', 'five')
```

```
In [28]: mytuple[:3]
```

```
Out[28]: ('one', 'two', 'three')
```

```
In [29]: mytuple[:2]
```

```
Out[29]: ('one', 'two')
```

```
In [30]: mytuple[-3:]
```

```
Out[30]: ('seven', 'eight', 'nine')
```

```
In [31]: mytuple[-2:]
```

```
Out[31]: ('eight', 'nine')
```

```
In [32]: mytuple[-1]
```

```
Out[32]: 'nine'
```

```
In [33]: mytuple[:] # Return whole the tuple
```

```
Out[33]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
```

```
In [34]: mytuple[::-1] #print reverse all the tuples
```

```
Out[34]: ('nine', 'eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one')
```

```
In [35]: mytuple[::-2] #Print every 2nd rverse index
```

```
Out[35]: ('nine', 'seven', 'five', 'three', 'one')
```

```
In [36]: mytuple
```

```
Out[36]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
```

Remove & change items

```
In [39]: mytuple
```

```
Out[39]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
```

```
In [40]: del mytuple[0] #tuples are immutable which means we canot DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 del mytuple[0]

TypeError: 'tuple' object doesn't support item deletion
```

In [41]:
```
del mytuple[100]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 1
----> 1 del mytuple[100]

TypeError: 'tuple' object doesn't support item deletion
```

In [42]:
```
mytuple[0]=1 #Tuples are immutable which means canot CHANGE tuples items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[42], line 1
----> 1 mytuple[0]=1 #

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

In [43]:
```
del mytuple # Deleting entire tuple object is possible
```

Loop through a tuple

In [46]:
```
mytuple=('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [47]:
```
for i in mytuple:
    print(i)
```

```
one
two
three
four
five
six
seven
eight
```

In [48]:
```
for i in enumerate(mytuple):print(i)
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
(7, 'eight')
```

Tuple Membership

In [49]:
```
mytuple
```

Out[49]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [50]:
```
'one' in mytuple #check if 'one' exist in the list
```

Out[50]:
```
True
```

In [51]:
```
'eleven' in mytuple #check if 'eleven' exist in the list
```

Out[51]:
```
False
```

In [52]:
```
'twelve' not in mytuple
```

Out[52]:
```
True
```

In [62]:
```
if 'three' in mytuple:
    print('three is present in the tuple')
else:
    print('three is not present in the tuple')
```

```
three is present in the tuple
```

In [66]:
```
if 'eleven' in mytuple:
    print('eleven is present in the tuple')
else:
    print('eleven is not present in the tuple')
```

```
eleven is not present in the tuple
```

In [68]:
```
if 'ten' in mytuple:
    print('ten is present in the tuple')
else:
    print('ten is not present in the tuple')
```

```
ten is not present in the tuple
```

index position

In [69]:
```
mytuple
```

```
Out[69]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [70]: mytuple.index('one')
```

```
Out[70]: 0
```

```
In [71]: mytuple.index('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[71], line 1
----> 1 mytuple.index(      )

ValueError: tuple.index(x): x not in tuple
```

```
In [72]: mytuple.index('two')
```

```
Out[72]: 1
```

```
In [73]: mytuple.index('three')
```

```
Out[73]: 2
```

```
In [74]: mytuple.index('four')
```

```
Out[74]: 3
```

```
In [75]: mytuple.index('five')
```

```
Out[75]: 4
```

sorting

```
In [76]: mytuple2=(10,20,30,40,50,60)
         print(mytuple2)
```

```
(10, 20, 30, 40, 50, 60)
```

```
In [77]: sorted(mytuple2) #Retirn a new sorted list and doesnot change orginal tuple
```

```
Out[77]: [10, 20, 30, 40, 50, 60]
```

```
In [78]: sorted(mytuple2,reverse=True) #sorted in descending order
```

```
Out[78]: [60, 50, 40, 30, 20, 10]
```

```
In [79]: mytuple3=(1,2,3,4,5,6,7,8,9,0)
         print(mytuple3)
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
```

```
In [80]: sorted(mytuple3)
```

Out[80]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [81]:
```
sorted(mytuple3,reverse=True)
```

Out[81]:
```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

In [ ]:
```

```