

## tuple-set-task

March 8, 2025

```
[1]: t = ()
```

```
[3]: t
```

```
[3]: ()
```

```
[5]: print(type(t))
```

```
<class 'tuple'>
```

```
[11]: t1=(10,20,30,40,50)
```

```
[17]: len(t1)
```

```
[17]: 5
```

```
[21]: t1.count(10)
```

```
[21]: 1
```

```
[23]: t1.count(40)
```

```
[23]: 1
```

```
[25]: t1.index(20)
```

```
[25]: 1
```

```
[27]: l5=['a','b','c','d']
```

```
[29]: l5
```

```
[29]: ['a', 'b', 'c', 'd']
```

```
[31]: l5[1]=10
```

```
[33]: l5
```

```
[33]: ['a', 10, 'c', 'd']
```

```
[35]: t2=(100,3.4,'nit',True,1+2j,[1,2,3],(5,6,7))
```

```
[39]: print(t1)
      print(t)
      print(t2)
```

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

```
()
```

```
(100, 3.4, 'nit', True, (1+2j), [1, 2, 3], (5, 6, 7))
```

```
[41]: t1[0]
```

```
[41]: 10
```

```
[43]: t1[0]=10
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[43], line 1
----> 1 t1[0]=10

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

```
[45]: icici = (46578, 'abcd75653', 235546, 648532)
      icici
```

```
[45]: (46578, 'abcd75653', 235546, 648532)
```

```
[47]: icici[0]= 1234
      icici
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[47], line 1
----> 1 icici[0]= 1234
      2 icici

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

```
[ ]: #Data stored in bank, details stored in tuple data structures
```

```
[49]: t1
```

```
[49]: (10, 20, 30, 40, 50)
```

```
[53]: t4=t1*3  
      t4
```

```
[53]: (10, 20, 30, 40, 50, 10, 20, 30, 40, 50, 10, 20, 30, 40, 50)
```

```
[55]: t4[:]
```

```
[55]: (10, 20, 30, 40, 50, 10, 20, 30, 40, 50, 10, 20, 30, 40, 50)
```

```
[58]: t1
```

```
[58]: (10, 20, 30, 40, 50)
```

```
[61]: t1[:7]
```

```
[61]: (10, 20, 30, 40, 50)
```

```
[63]: t1
```

```
[63]: (10, 20, 30, 40, 50)
```

```
[65]: t1[0:10:2]
```

```
[65]: (10, 30, 50)
```

```
[67]: t1.remove(10)
```

```
-----  
AttributeError                                Traceback (most recent call last)  
Cell In[67], line 1  
----> 1 t1.remove(10)  
  
AttributeError: 'tuple' object has no attribute 'remove'
```

```
[69]: t2
```

```
[69]: (100, 3.4, 'nit', True, (1+2j), [1, 2, 3], (5, 6, 7))
```

```
[73]: t2.index(100)
```

```
[73]: 0
```

```
[75]: t2.index('nit')
```

```
[75]: 2
```

```
[77]: x=5  
      y=2  
      x//y
```

```
[77]: 2
```

```
[79]: x=5  
      y=2  
      x/y
```

```
[79]: 2.5
```

```
[81]: x=5  
      y=2  
      x//y  
      x/y
```

```
[81]: 2.5
```

## 1 tuple is immutable data type

## 2 Set

```
[2]: s={}
```

```
[4]: type(s)
```

```
[4]: dict
```

```
[6]: s=()
```

```
[8]: type(s)
```

```
[8]: tuple
```

```
[10]: s1={10,20}
```

```
[12]: type(s1)
```

```
[12]: set
```

```
[20]: s2=set() # empty st defining
```

```
[16]: type(s2)
```

```
[16]: set
```

```
[22]: s2={20,100,30,45} # set represent ordered set
```

```
[24]: s4={1,2,3,'nit',1+2j,[1,2,3],[2,5,6],True}
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[24], line 1  
----> 1 s4={1,2,3,'nit',1+2j,[1,2,3],[2,5,6],True}  
  
TypeError: unhashable type: 'list'
```

```
[27]: s5={2,3.4,'nt',False,1+2j}
```

```
[33]: print(s1)  
      print(s2)  
      print(s5)
```

```
{10, 20}  
{45, 100, 20, 30}  
{False, 2, 3.4, (1+2j), 'nt'}
```

```
[31]: del s4
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[31], line 1  
----> 1 del s4  
  
NameError: name 's4' is not defined
```

```
[41]: s2
```

```
[41]: {20, 30, 45, 100}
```

```
[45]: s2.add(3)
```

```
[47]: s2
```

```
[47]: {3, 20, 30, 45, 100}
```

```
[49]: s2.add(200)  
      s2
```

```
[49]: {3, 20, 30, 45, 100, 200}
```

```
[51]: s1[1:4] #Indexing and slicing is not allowed
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[51], line 1  
----> 1 s1[1:4]  
  
TypeError: 'set' object is not subscriptable
```

```
[53]: s5
```

```
[53]: {(1+2j), 2, 3.4, False, 'nt'}
```

```
[58]: s4=s5.copy()
```

```
[60]: s4
```

```
[60]: {(1+2j), 2, 3.4, False, 'nt'}
```

```
[67]: s4.add(2) # duplicate is not allowed
```

```
[65]: s4
```

```
[65]: {(1+2j), 2, 3.4, False, 'nt'}
```

```
[69]: s5.clear()
```

```
[71]: s5
```

```
[71]: set()
```

```
[73]: del s5
```

```
[75]: s4
```

```
[75]: {(1+2j), 2, 3.4, False, 'nt'}
```

```
[79]: s4.remove((1+2j))
```

```
[82]: s4
```

```
[82]: {2, 3.4, False, 'nt'}
```

```
[84]: s4.remove(2,3.4) #delete only takes one argument
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[84], line 1
```

```
----> 1 s4.remove(2,3.4)
```

```
TypeError: set.remove() takes exactly one argument (2 given)
```

```
[88]: s3 = {'c','e','l', 'f','j'}
```

```
[91]: s3.discard('m')
```

```
[93]: s3
```

```
[93]: {'c', 'e', 'f', 'j', 'l'}
```

```
[95]: s3.remove('m')
```

```
-----  
KeyError                                Traceback (most recent call last)  
Cell In[95], line 1  
----> 1 s3.remove('m')  
  
KeyError: 'm'
```

```
[97]: s3
```

```
[97]: {'c', 'e', 'f', 'j', 'l'}
```

```
[99]: s3.discard('f') #never give you the error if element is not present
```

```
[101]: s3
```

```
[101]: {'c', 'e', 'j', 'l'}
```

```
[103]: s3.pop()
```

```
[103]: 'j'
```

```
[105]: s2
```

```
[105]: {3, 20, 30, 45, 100, 200}
```

```
[107]: s2.pop(3) #indexing not allowed
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[107], line 1  
----> 1 s2.pop(3)
```

```
TypeError: set.pop() takes no arguments (1 given)
```

```
[124]: s4
```

```
[124]: {2, 3.4, False, 'nt'}
```

```
[127]: for i in s3:  
        print(i)
```

```
l  
c  
e
```

```
[131]: for i in enumerate(s3): # shows the index  
        print(i)
```

```
(0, 'l')  
(1, 'c')  
(2, 'e')
```

```
[133]: 5 in s2
```

```
[133]: False
```

```
[135]: s2
```

```
[135]: {3, 20, 30, 45, 100, 200}
```

```
[137]: 45 in s2
```

```
[137]: True
```

```
[139]: s2.add(300)
```

```
[142]: s2.update(s3)
```

```
[144]: s2
```

```
[144]: {100, 20, 200, 3, 30, 300, 45, 'c', 'e', 'l'}
```

```
[146]: s3
```

```
[146]: {'c', 'e', 'l'}
```

```
[148]: s2
```



```
[148]: {100, 20, 200, 3, 30, 300, 45, 'c', 'e', 'l'}
```

### 3 Set Operation

```
[153]: s6={1,2,3,4,5}  
      s7={4,5,6,7,8}  
      s8={8,9,10}
```

```
[157]: s6.union(s7)
```

```
[157]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
[161]: s6.union(s7,s8) # allows more than one argument
```

```
[161]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
[175]: s6 | s7 # for union using pipr |
```

```
[175]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
[177]: s6|s7|s8
```

```
[177]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
[179]: s6
```

```
[179]: {1, 2, 3, 4, 5}
```

```
[182]: s7
```

```
[182]: {4, 5, 6, 7, 8}
```

```
[184]: s8
```

```
[184]: {8, 9, 10}
```

```
[186]: s6.intersection(s7)
```

```
[186]: {4, 5}
```

```
[188]: s7.intersection(s8)
```

```
[188]: {8}
```

```
[190]: s6 & s7 # for intersection
```

[190]: {4, 5}

```
[192]: print(s6)
      print(s7)
      print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
[194]: s6.difference(s7)
```

[194]: {1, 2, 3}

```
[196]: s6-s7
```

[196]: {1, 2, 3}

```
[201]: s7-s8
```

[201]: {4, 5, 6, 7}

```
[203]: print(s6)
      print(s7)
      print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
[205]: s8-s7
```

[205]: {9, 10}

```
[207]: print(s6)
      print(s7)
      print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
[241]: s6.symmetric_difference(s7) # different element from both sets
```

[241]: {1, 2, 3, 6, 7, 8}

```
[ ]:
```



[ ]:

[ ]:

[ ]:

[ ]: