

24-10-2025

tuple

```
In [2]: t = ()  
t
```

```
Out[2]: ()
```

```
In [3]: type(t)
```

```
Out[3]: tuple
```

```
In [4]: t = (10,20,30)  
t
```

```
Out[4]: (10, 20, 30)
```

```
In [5]: t.count(10)
```

```
Out[5]: 1
```

```
In [6]: t.count(20)
```

```
Out[6]: 1
```

```
In [7]: t1 = (10,20,2.2,'ten',True,1+2j)  
t1
```

```
Out[7]: (10, 20, 2.2, 'ten', True, (1+2j))
```

```
In [8]: t1.count(20)
```

```
Out[8]: 1
```

```
In [9]: t1 = (10,20,2.2,'ten',True,1+2j,20)  
t1
```

```
Out[9]: (10, 20, 2.2, 'ten', True, (1+2j), 20)
```

```
In [10]: t1.count(20)
```

```
Out[10]: 2
```

```
In [11]: t1.index(20)
```

```
Out[11]: 1
```

```
In [12]: print(t)
```

```
print(t1)
```

```
(10, 20, 30)
(10, 20, 2.2, 'ten', True, (1+2j), 20)
```

```
In [13]: print(len(t))
         print(len(t1))
```

```
3
7
```

```
In [14]: t
```

```
Out[14]: (10, 20, 30)
```

```
In [15]: t[0]
```

```
Out[15]: 10
```

```
In [16]: t[0] = 100
```

```
-----
TypeError                                 Traceback (most recent call last)
Cell In[16], line 1
----> 1 t[0] = 100
```

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

```
In [17]: bank_account = (1234, 'cizp45yi',10000)
         bank_account
```

```
Out[17]: (1234, 'cizp45yi', 10000)
```

```
In [18]: bank_account[2] = 200000
```

```
-----
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 bank_account[2] = 200000
```

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

```
In [19]: bank_account.
```

```
Cell In[19], line 1
    bank_account.
               ^
SyntaxError: invalid syntax
```

```
In [20]: t
```

```
Out[20]: (10, 20, 30)
```

```
In [21]: t2 = t * 3
         t2
```

```
Out[21]: (10, 20, 30, 10, 20, 30, 10, 20, 30)
```

```
In [22]: t
```

```
Out[22]: (10, 20, 30)
```

```
In [23]: for i in t:  
         print(i)
```

```
10
```

```
20
```

```
30
```

```
In [ ]:
```

SET

```
In [24]: s = {}  
s
```

```
Out[24]: {}
```

```
In [25]: type(s)
```

```
Out[25]: dict
```

```
In [26]: s1 = set()  
s1
```

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

```
In [27]: s2 = {90, 10, 50, 40, 25, 10, 50}  
s2
```

```
Out[27]: {10, 25, 40, 50, 90}
```

```
In [28]: type(s2)
```

```
Out[28]: set
```

```
In [29]: s2
```

```
Out[29]: {10, 25, 40, 50, 90}
```

```
In [30]: s3 = s2.copy()  
s3
```

```
Out[30]: {10, 25, 40, 50, 90}
```

```
In [31]: s3
```

```
Out[31]: {10, 25, 40, 50, 90}
```

```
In [32]: s3.add(3.4)
```

```
In [33]: s3
```

```
Out[33]: {3.4, 10, 25, 40, 50, 90}
```

```
In [34]: s3.add('nit')
```

```
In [35]: s3
```

```
Out[35]: {10, 25, 3.4, 40, 50, 90, 'nit'}
```

```
In [36]: s3.add(1+2j)
s3.add(True)
```

```
In [37]: s3
```

```
Out[37]: {(1+2j), 10, 25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [38]: print(s)
print(s1)
print(s2)
print(s3)
```

```
{  
set(  
{50, 90, 40, 25, 10}  
{True, 3.4, (1+2j), 10, 'nit', 25, 90, 40, 50}
```

```
In [39]: s
```

```
Out[39]: {}
```

```
In [40]: type(s)
```

```
Out[40]: dict
```

```
In [41]: s2
```

```
Out[41]: {10, 25, 40, 50, 90}
```

```
In [42]: s3
```

```
Out[42]: {(1+2j), 10, 25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [43]: s3.remove(2000)
```

KeyError

Traceback (most recent call last)

Cell In[43], line 1

----> 1 s3.remove(2000)

KeyError: 2000

In [44]: s3.remove(1+2j)

In [45]: s3

Out[45]: {10, 25, 3.4, 40, 50, 90, True, 'nit'}

In [46]: s3

Out[46]: {10, 25, 3.4, 40, 50, 90, True, 'nit'}

In [47]: s3.discard(10)

In [48]: s3

Out[48]: {25, 3.4, 40, 50, 90, True, 'nit'}

In [49]: s3.discard(2000)

In [50]: s3

Out[50]: {25, 3.4, 40, 50, 90, True, 'nit'}

In [51]: s3.pop()

Out[51]: True

In [52]: s3

Out[52]: {25, 3.4, 40, 50, 90, 'nit'}

In [53]: s3.pop()

Out[53]: 3.4

In [54]: s3

Out[54]: {25, 40, 50, 90, 'nit'}

In [55]: s3.pop(0)

TypeError

Traceback (most recent call last)

Cell In[55], line 1

----> 1 s3.pop(0)

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

In [56]: `s3[:]`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[56], line 1  
----> 1 s3[:]  
  
TypeError: 'set' object is not subscriptable
```

In [57]: `s3`

Out[57]: {25, 40, 50, 90, 'nit'}

In [58]: `s3[1:]`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[58], line 1  
----> 1 s3[1:]  
  
TypeError: 'set' object is not subscriptable
```

In [59]: `s3`

Out[59]: {25, 40, 50, 90, 'nit'}

In [60]: `s3`

Out[60]: {25, 40, 50, 90, 'nit'}

In [61]: `s3.pop(0)`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[61], line 1  
----> 1 s3.pop(0)  
  
TypeError: set.pop() takes no arguments (1 given)
```

In [62]: `s3.pop()`

Out[62]: 'nit'

In [63]: `s3`

Out[63]: {25, 40, 50, 90}

In [64]: `40 in s3`

Out[64]: True

SET OPERATIONS

```
In [65]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
```

```
In [66]: type(c)
```

```
Out[66]: set
```

```
In [67]: a.union(b)
```

```
Out[67]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [68]: a.union(b, c)
```

```
Out[68]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [69]: print(a)
         print(b)
         print(c)
```

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

```
In [70]: a | b
```

```
Out[70]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [71]: b | c
```

```
Out[71]: {4, 5, 6, 7, 8, 9, 10}
```

```
In [72]: a | b | c
```

```
Out[72]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [73]: a | c
```

```
Out[73]: {1, 2, 3, 4, 5, 8, 9, 10}
```

```
In [74]: a | c | b
```

```
Out[74]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Intersection-common items

```
In [75]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
```

```
In [76]: a.intersection(b)
```

Out[76]: {4, 5}

In [77]: `b.intersection(c)`

Out[77]: {8}

In [78]: `a & b`

Out[78]: {4, 5}

In [79]: `b & c`

Out[79]: {8}

Difference

In [80]: `a = {1,2,3,4,5}`
`b = {4,5,6,7,8}`
`c = {8,9,10}`

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

Out[81]: {1, 2, 3}

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

Out[82]: {6, 7, 8}

In [83]: `b.difference(c)`

Out[83]: {4, 5, 6, 7}

In [84]: `b - c`

Out[84]: {4, 5, 6, 7}

In [85]: `c - b`

Out[85]: {9, 10}

In [86]: `a - b - c`

Out[86]: {1, 2, 3}

In []:

27 October 2025

In [87]: `print(a)`
`print(b)`


```
print(c)
```

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

```
In [88]: a.symmetric_difference(b)
```

```
Out[88]: {1, 2, 3, 6, 7, 8}
```

```
In [89]: b^c
```

```
Out[89]: {4, 5, 6, 7, 9, 10}
```

```
In [90]: print(a)  
print(b)  
print(c)
```

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

```
In [91]: a^c
```

```
Out[91]: {1, 2, 3, 4, 5, 8, 9, 10}
```

```
In [92]: print(a)  
print(b)  
print(c)
```

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

```
In [93]: a.symmetric_difference_update(b)
```

```
In [94]: a
```

```
Out[94]: {1, 2, 3, 6, 7, 8}
```

```
In [95]: print(a)  
print(b)  
print(c)
```

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

superset, subset, disjoint

```
In [96]: s4 = {1,2,3,4,5,6,7,8,9}  
s5 = {3,4,5,6,7,8}  
s6 = {10,20,30,40}
```

```
In [97]: s4.issuperset(s5)
```

```
Out[97]: True
```

```
In [98]: s5.issubset(s4)
```

```
Out[98]: True
```

```
In [99]: s6.isdisjoint(s5)
```

```
Out[99]: True
```

```
In [100... s4 = {1,2,3,4,5,6,7,8,9}
s5 = {3,4,5,6,7,8}
s6 = {10,20,30,40}
```

```
In [101... s6.issubset(s5)
```

```
Out[101... False
```

```
In [102... s6.issubset(s4)
```

```
Out[102... False
```

```
In [103... s7 = {1,2,3,4,5,6,7,8,9}
s8 = {15,25,35}
s9 = {10,20,30,40}
```

```
In [104... s7.issuperset(s8)
```

```
Out[104... False
```

```
In [109... s8.issubset(s7)
```

```
Out[109... False
```

```
In [111... s7.isdisjoint(s8)
```

```
Out[111... False
```

Dictionary

```
In [134... d = {}
d
```

```
Out[134... {}
```

```
In [135... type(d)
```

```
Out[135... dict
```

```
In [136... d1 = {1 : 'one', 2: 'two', 3: 'three', 'four': 4, 'l' : [1,2,3]}
d1
```

```
Out[136... {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [137... d1
```

```
Out[137... {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [138... d2 = d1.copy()
```

```
In [139... d2
```

```
Out[139... {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [140... d1.items()
```

```
Out[140... dict_items([(1, 'one'), (2, 'two'), (3, 'three'), ('four', 4), ('l', [1, 2, 3])])
```

```
In [141... len(d1.items())
```

```
Out[141... 5
```

```
In [142... d1
```

```
Out[142... {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [143... d1[:]
```

KeyError

Traceback (most recent call last)

Cell In[143], line 1

----> 1 d1[:]

KeyError: slice(None, None, None)

```
In [144... d1
```

```
Out[144... {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [145... d1['four']
```

```
Out[145... 4
```

```
In [146... d1['one']
```

KeyError

Traceback (most recent call last)

Cell In[146], line 1

----> 1 d1['one']

KeyError: 'one'

```
In [147... d1
```

```
Out[147...] {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [148...] d1.keys()
```

```
Out[148...] dict_keys([1, 2, 3, 'four', 'l'])
```

```
In [149...] d1.values()
```

```
Out[149...] dict_values(['one', 'two', 'three', 4, [1, 2, 3]])
```

```
In [150...] d1
```

```
Out[150...] {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [151...] d1.values()
```

```
Out[151...] dict_values(['one', 'two', 'three', 4, [1, 2, 3]])
```

```
In [152...] d1
```

```
Out[152...] {1: 'one', 2: 'two', 3: 'three', 'four': 4, 'l': [1, 2, 3]}
```

```
In [153...] d1.pop('l')
```

```
Out[153...] [1, 2, 3]
```

```
In [154...] d1
```

```
Out[154...] {1: 'one', 2: 'two', 3: 'three', 'four': 4}
```

```
In [155...] 100 in d1
```

```
Out[155...] False
```

Range

```
In [156...] range(10)
```

```
Out[156...] range(0, 10)
```

```
In [157...] range(10,20)
```

```
Out[157...] range(10, 20)
```

```
In [158...] range(10,20,5)
```

```
Out[158...] range(10, 20, 5)
```

```
In [159...] list(range(10))
```

```
Out[159...] [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [160... list(range(10,20))
```

```
Out[160... [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
In [161... list(range(10,20,5))
```

```
Out[161... [10, 15]
```

```
In [163... r = range(10,20,5)
r
```

```
Out[163... range(10, 20, 5)
```

```
In [164... for i in r:
    print(i)
```

```
10
```

```
15
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]: #List
mutable
duplicate is allowed
append(),copy(),insert(),extend(),pop(),
remove the element
list is growable
multiple data type in a list
indexing & slicing is allowed

#Tuple
immutable(unchangeable)
duplication is allowed
remove is not allowed
only 2 function will work(.index,.count)
```

```
In [ ]:
```