

## #Tuple in python

```
# 4 Aug
#Tuple - always assigned with () brackets
#Tuple is immutable - once assigned we can not change the value

t = ()
t
```

[24] ✓ 0.0s Python

( )

```
type(t)
```

[25] ✓ 0.0s Python

tuple

```
t = (10,20,30)
t
```

[26] ✓ 0.0s Python

(10, 20, 30)

```
len(t)
```

[27] ✓ 0.0s Python

3

```
t1 = (10,2.3, "nit",1+2j,True, 10,20)
t1
```

[28] ✓ 0.0s Python

(10, 2.3, 'nit', (1+2j), True, 10, 20)

```
print(t)
print(t1)
```

[29] ✓ 0.0s Python

(10, 20, 30)  
(10, 2.3, 'nit', (1+2j), True, 10, 20)

```
t.count(10)
```

[30] ✓ 0.0s Python

1

```
t1.count(10)
```

[31] ✓ 0.0s Python

2

```
t1.count(2.3)
```

[32] ✓ 0.0s Python

1

```
[33] ✓ 0.0s Python
... 1

t1.index(2.3)

t
[34] ✓ 0.0s Python
... (10, 20, 30)

sbi = (123456, "yryfjj4867", 'dob-3rd march 1987', 'mob-123456')
sbi
[35] ✓ 0.0s Python
... (123456, 'yryfjj4867', 'dob-3rd march 1987', 'mob-123456')
Generate + Code + Markdown

sbi[0]= 857684 #not change (read only mode)
[36] ⊗ 0.1s Python
...
-----
TypeError                                Traceback (most recent call)
Cell In[36], line 1
----> 1 sbi[0]= 857684

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

```
t
[37] ✓ 0.0s Python
... (10, 20, 30)

t2 =t * 3
[38] ✓ 0.0s Python
...
✓ t2 ...

(10, 20, 30, 10, 20, 30, 10, 20, 30)

print(t)
print(t1)
print(t2)
[40] ✓ 0.0s Python
... (10, 20, 30)
(10, 2.3, 'nit', (1+2j), True, 10, 20)
(10, 20, 30, 10, 20, 30, 10, 20, 30)

t1[1:6]
[41] ✓ 0.0s Python
... (2.3, 'nit', (1+2j), True, 10)
```

```
t2[1:6:3] # print the index value starting from 1st index ends wit
[42] ✓ 0.0s Python
... (20, 20)

t2[1:6:5]
[43] ✓ 0.0s Python
... (20,)

t1[:]
[44] ✓ 0.0s Python
... (10, 2.3, 'nit', (1+2j), True, 10, 20)

t1[::-1]
[46] ✓ 0.0s Python
... (20, 10, True, (1+2j), 'nit', 2.3, 10)

t2
[47] ✓ 0.0s Python
... (10, 20, 30, 10, 20, 30, 10, 20, 30)
```

```
t2[::-2]
[48] ✓ 0.0s Python
... (30, 10, 20, 30, 10)

t2[::2]
[49] ✓ 0.0s Python
... (10, 30, 20, 10, 30)

t2[::3]
[50] ✓ 0.0s Python
... (30, 30, 30)

t3 = (1,2,3,4,5,6,7,8)
print(t3[:])
print(t3[1:9])
print(t3[1:])
print(t3[:9])
print(t3[1:9:3])
print(t3[::-1])
print(t3[::])
print(t3[::3])
print(t3[3:-1])
[53] ✓ 0.0s Python
```

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

```
print(t3[1:7:3])
print(t3[::-3])
```

```
[55] ✓ 0.0s
```

Python

```
... (2, 5)
      (8, 5, 2)
```

```
for i in t3:
    print(i)
```

```
[57] ✓ 0.0s
```

Python

```
... 1
      2
      3
      4
      5
      6
      7
      8
```

```
▶ for i in enumerate(t1):
    | print(i)
```

```
[58] ✓ 0.0s
```

Python

```
... (0, 10)
      (1, 2.3)
      (2, 'nit')
      (3, (1+2j))
      (4, True)
      (5, 10)
      (6, 20)
```

```
#Range
range(5)# this arg u can pass at most 3 argument(start stop and st
```

```
[59] ✓ 0.0s
```

Python

```
... range(0, 5)
```

```
list(range(5))
```

```
[60] ✓ 0.0s
```

Python

```
... [0, 1, 2, 3, 4]
```

```
list(range(10,20))
```

```
[61] ✓ 0.0s
```

Python

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

```
▶ list(range(10,50,5))
```

```
[63] ✓ 0.0s
```

Python

```
... [10, 15, 20, 25, 30, 35, 40, 45]
```