Tuple

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

Out[1]: ()

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

Out[2]: tuple

```
In [3]: t1=(10,20,30)
```

```
In [4]: t1
```

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

```
In [6]: t1.count(20)
```

Out[6]: 1

```
In [7]: t1.index(10)
```

Out[7]: 0

```
In [8]: for i in t1:
            print(i)
```

```
10
20
30
```

```
In [10]: for i in enumerate(t1):
             print(i)
```

```
(0, 10)
(1, 20)
(2, 30)
```

```
In [11]: t2=t1*3
```

```
In [12]: t2
```

Out[12]: (10, 20, 30, 10, 20, 30, 10, 20, 30)

```
In [13]: type(t2)
```

Out[13]: tuple

```
In [14]: del t2
```

```
In [15]: del t1
```

```
In [17]: t1=(10,20,30)
```

```
In [18]: t2=(11.11,22.22,33.33)
```

```
In [19]: t3=('one','two','three')
```

```
In [20]: t4=(10,'nit',22.2,(10,20),(30,40))
```

```
In [21]: len(t4)
```

Out[21]: 5

```
In [22]: t4[3]
```

Out[22]: (10, 20)

```
In [23]: t4[3][1]
```

Out[23]: 20

```
In [25]: t4[-1]
```

Out[25]: (30, 40)

```
In [27]: t5=('one','two','three','four','five','six','seven','eight')
```

```
In [28]: t5[0:3]
```

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

```
In [29]: t5[2:5]
```

Out[29]: ('three', 'four', 'five')

```
In [30]: t5[:3]
```

Out[30]: ('one', 'two', 'three')

```
In [31]: t5[:2]
```

Out[31]: ('one', 'two')

```
In [32]: t5[-3:]
```

Out[32]: ('six', 'seven', 'eight')

```
In [33]: t5[-2]
```

Out[33]: 'seven'

In [34]: ```python
t5[-2:]
```

Out[34]: ('seven', 'eight')

In [35]: ```python
t5[:]
```

Out[35]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [36]: ```python
del t5[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[36], line 1
----> 1 del t5[1]

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

In [37]: ```python
t5[3]=4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[37], line 1
----> 1 t5[3]=4

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

In [38]: ```python
del t5
```

In [39]: ```python
t5
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[39], line 1
----> 1 t5

NameError: name 't5' is not defined
```

In [40]: ```python
t5=('one','two','three','four','five','six','seven','eight')
```

In [43]: ```python
for i in t5:
    print(i)
```

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

In [44]: ```python
for i in enumerate(t5):
    print(i)
```

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

In [45]: `'one' in t5`

Out[45]: True

In [46]: `'ten' in t5`

Out[46]: False

In [47]:
```python
if 'four' in t5:
    print('four present in tuple')
else:
    print('four is not present in tuple')
```

four present in tuple

In [48]:
```python
if 'ten' in t5:
    print('ten present in tuple')
else:
    print('ten is not present in tuple')
```

ten is not present in tuple

In [49]: `t5.index('four')`

Out[49]: 3

In [50]: `t5.index('ten')`

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[50], line 1
----> 1 t5.index('ten')

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

In [51]: `t6=(2,6,3,6,9,1,4)`

In [53]: `sorted(t6)`

Out[53]: [1, 2, 3, 4, 6, 6, 9]

In [54]: `t6`

Out[54]: (2, 6, 3, 6, 9, 1, 4)

In [55]: `sorted(t6,reverse=True)`

Out[55]: [9, 6, 6, 4, 3, 2, 1]

In [61]:
```python
a=6
isinstance(a,int)
```

Out[61]: True

In [64]:
```python
str1="hello"
print(str1[1:4])
```

ell

In [67]:
```python
x= [1, 2, 3, 4]
y = filter(lambda a: a % 2 == 0, x)
print(list(y))
```

[2, 4]

In [74]:
```python
print(str(True), end=" ")
int("4.5")
```

True
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[74], line 2
      1 print(str(True), end=" ")
----> 2 int("4.5")

ValueError: invalid literal for int() with base 10: '4.5'
```

In [ ]:

In [ ]:

In [ ]: