# Tuple Creation

In [1]:
```
tup = ()   # empty tuple
```

In [2]:
```
tup
```

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

In [3]:
```
tup2=(10,30,60)      # integer tuple
```

In [4]:
```
tup3=(10.77,30.66,60.89)    # float tuple
```

In [5]:
```
tup4=('one','two','three')  # string tuple
```

In [15]:
```
tup5=('nit',25,(50,100), 4.58,(150,'arif'))  # nested tuple
```

In [16]:
```
tup6 = ('nit',5.89,894,1+7j, True)   # mixed datatype tuple
```

In [17]:
```
tup7 = ('nit',25,[50,100],[150,90],{'john','david'},(999,22,33))
```

In [18]:
```
len(tup7)
```

Out[18]:
```
6
```

In [ ]:
```

```

# Tuple indexing

In [19]:
```
tup2[0]
```

Out[19]:
```
10
```

In [21]:
```
tup5[4][1]
```

Out[21]:
```
'arif'
```

In [25]:
```
tup7[4][1]

# set is unordered so in order to call it, it should be converted to a list or tupl
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 tup7[4][1]

TypeError: 'set' object is not subscriptable
```

In [26]:
```
tup7
```

Out[26]:
```
('nit', 25, [50, 100], [150, 90], {'david', 'john'}, (999, 22, 33))
```

In [27]:
```
# converting set to list/tuple

tup7 = ('nit',25,[50,100],[150,90],('john','david'),(999,22,33))
tup7[4][1]
```

Out[27]:
```
'david'
```

In [28]:
```
tup7[-1]
```

Out[28]:
```
(999, 22, 33)
```

In [23]:
```
tup7[3][1]
```

Out[23]:
```
90
```

In [24]:
```
tup7[2][1]
```

Out[24]:
```
100
```

In [ ]:
```

```

# Tuple slicing

In [29]:
```
mytuple=('one','two','three','four','five','six','seven','eight')
```

In [30]:
```
mytuple[0:3]
```

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

In [31]:
```
mytuple[2:5]
```

Out[31]:
```
('three', 'four', 'five')
```

In [32]:
```
mytuple[:3]
```

Out[32]:
```
('one', 'two', 'three')
```

In [33]:
```
mytuple[3:]
```

Out[33]:
```
('four', 'five', 'six', 'seven', 'eight')
```

In [34]:
```
mytuple[:-2]
```

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

In [35]:
```
mytuple[-3:]
```

Out[35]:
```
('six', 'seven', 'eight')
```

In [36]:
```
mytuple[-4]
```

Out[36]:
```
'five'
```

In [38]:
```
mytuple[:]
```

Out[38]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [40]:
```
mytuple[::-1]
```

Out[40]:
```
('eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one')
```

In [ ]:

# Remove or change items

In [41]:
```
mytuple
```

Out[41]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [43]:
```
del mytuple[0]

# Tuples are immutable = cant delete existing items
```

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

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

In [44]:
```
mytuple[1] = 2

# Tuples are immutable = cant change existing items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 1
----> 1 mytuple[1] = 2

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

In [ ]:

# Loop through a tuple

In [45]:
```
mytuple
```

Out[45]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [46]:
```
for i in mytuple:
    print(i)
```

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

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

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

In [ ]:
```

```

# Tuple membership

In [48]:
```
mytuple
```

Out[48]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [49]:
```
'one' in mytuple
```

Out[49]:
```
True
```

In [50]:
```
'One' in mytuple   # different case string
```

Out[50]:
```
False
```

In [51]:
```
'ten' in mytuple
```

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

In [52]:
```
if 'three' in mytuple:
    print('Three is present in mytuple')
else:
    print('Three is not present in mytuple')
```

```
Three is present in mytuple
```

In [53]:
```
if 'ten' in mytuple:
    print('Ten is present in mytuple')
else:
    print('Ten is not present in mytuple')
```

```
Ten is not present in mytuple
```

In [ ]:
```

```

# Index position

In [54]:
```
mytuple
```

Out[54]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [55]:
```
mytuple.index('two')
```

Out[55]:
```
1
```

In [56]:
```
mytuple.index('five')
```

Out[56]:
```
4
```

In [59]:
```
mytuple.index('eight')
```

Out[59]:
```
7
```

In [ ]:
```

```

# Sorting

In [61]:
```
mytuple2=(49,12,786,148,90,37)
```

In [62]:
```
sorted(mytuple2)
```

Out[62]:
```
[12, 37, 49, 90, 148, 786]
```

In [63]:
```
sorted(mytuple2,reverse=True)
```

Out[63]:
```
[786, 148, 90, 49, 37, 12]
```

In [ ]:
```

```