```
In [2]:  #Tuple creation
         #Empty tuple
         tup1 = ()
         tup1

Out[2]:  ()
```

```
In [3]:  type(tup1)

Out[3]:  tuple
```

```
In [4]:  tup2 = (1,2,3)   #Tuple of integers
         tup2

Out[4]:  (1, 2, 3)
```

```
In [6]:  tup3 = (1.5, 3.2, 9.6)    #Tuple of float numbers
         tup3

Out[6]:  (1.5, 3.2, 9.6)
```

```
In [7]:  tup4 = ('ABC','XYZ','PQR')     #Tuple of string
         tup4

Out[7]:  ('ABC', 'XYZ', 'PQR')
```

```
In [8]:  tup5 = ('Rashmi',25,(10,20),(100,200))    #Nested tuples
         tup5

Out[8]:  ('Rashmi', 25, (10, 20), (100, 200))
```

```
In [9]:  tup6 = (100,'Rashmi',100.5)    #Tuple of mixed datatypes
         tup6

Out[9]:  (100, 'Rashmi', 100.5)
```

```
In [10]: tup7 = ('Rashmi',10,[20,40],[10,76],{'Sidhesh','Balurkar'},(99,22,33))
         tup7

Out[10]: ('Rashmi', 10, [20, 40], [10, 76], {'Balurkar', 'Sidhesh'}, (99, 22, 33))
```

```
In [11]: type(tup7)

Out[11]: tuple
```

```
In [12]: len(tup7)

Out[12]: 6
```

```
In [ ]:  #Tuple Indexing
In [13]: tup2[0]

Out[13]: 1

In [14]: tup4[0]

Out[14]: 'ABC'

In [15]: tup4[0][0]

Out[15]: 'A'

In [16]: tup4[-1]

Out[16]: 'PQR'

In [17]: tup5[-1]

Out[17]: (100, 200)

In [ ]: #Tuple Slicing

In [18]: mytuple = ('one','two','three','four','five','six','seven','eight')
         mytuple

Out[18]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [19]: mytuple[0:3]

Out[19]: ('one', 'two', 'three')

In [20]: mytuple[2:5]

Out[20]: ('three', 'four', 'five')

In [21]: mytuple[:3]

Out[21]: ('one', 'two', 'three')

In [22]: mytuple[:2]

Out[22]: ('one', 'two')

In [23]: mytuple[-3:]

Out[23]: ('six', 'seven', 'eight')

In [24]: mytuple[-2:]

Out[24]: ('seven', 'eight')
```

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

Out[25]:
```
'eight'
```

In [26]:
```
mytuple[:]
```

Out[26]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [27]:
```
#Remove and change items
mytuple
```

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

In [28]:
```
del mytuple[0]    #Tuples are immutable that means we can't DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 del mytuple[0]

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

In [29]:
```
mytuple[0] = 1       #Tuples are immutable that means we can't CHANGE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 mytuple[0] = 1

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

In [30]:
```
del mytuple    #Deleting entire tuple object is possible
```

In [31]:
```
mytuple
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[31], line 1
----> 1 mytuple

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

In [33]:
```
#Loop through tuple
```

In [34]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
mytuple
```

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

In [35]:
```
for i in mytuple:
    print(i)
```

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

```
In [36]: for i in enumerate(mytuple):
             print(i)
```

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

```
In [37]: #Tuple membership
         mytuple
```

```
Out[37]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [38]: 'one' in mytuple    #Check if 'one' exist in the tuple items
```

```
Out[38]: True
```

```
In [39]: 'ten' in mytuple     #Check if 'ten' exist in the tuple items
```

```
Out[39]: False
```

```
In [40]: if 'three' in mytuple:                         #Check if 'three' exist in the tu
             print('Three is present in the tuple')
         else:
             print('Three is not present in the tuple')
```

```
Three is present in the tuple
```

```
In [42]: if 'eleven' in mytuple:                         #Check if 'eleven' exist in the
             print('Eleven is present in the tuple')
         else:
             print('Eleven is not present in the tuple')
```

```
Eleven is not present in the tuple
```

```
In [43]: #Index position
         mytuple
```

```
Out[43]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [45]: mytuple.index('one')
```

```
Out[45]: 0
```

```
In [46]: mytuple.index('five')
```

```
Out[46]: 4
```

```
In [51]: mytuple1 = mytuple
         mytuple1
```

```
Out[51]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [52]: mytuple1.index('one')
```

```
Out[52]: 0
```

```
In [53]: #Sorting
         mytuple2 = (43,67,99,12,6,90,67)
         mytuple2
```

```
Out[53]: (43, 67, 99, 12, 6, 90, 67)
```

```
In [54]: sorted(mytuple2)      #Returns a new sorted tuple and doesn't change the original t
```

```
Out[54]: [6, 12, 43, 67, 67, 90, 99]
```

```
In [55]: sorted(mytuple2, reverse='True')      #Sort in descending order
```

```
Out[55]: [99, 90, 67, 67, 43, 12, 6]
```

```
In [56]: x=5
         type(x) == "int"
```

```
Out[56]: False
```

```
In [57]: type(x)=int
```

```
  Cell In[57], line 1
    type(x)=int
    ^
SyntaxError: cannot assign to function call here. Maybe you meant '==' instead of
'='?
```

```
In [58]: isinstance(x,int)
```

```
Out[58]: True
```

```
In [59]: s = ['ABC','PQR','HYT']     #Convert the list of strings to a single string
```

```
In [61]: s
```

```
Out[61]: ['ABC', 'PQR', 'HYT']
```

```
In [62]: str(s)

Out[62]: "['ABC', 'PQR', 'HYT']"
```

```
In [63]: print(type({}))

<class 'dict'>
```

```
In [64]: var = 2
         print(2==2.0)

True
```

```
In [65]: num = 4+0j
         print(type(num))

<class 'complex'>
```

```
In [66]: print(int(3.9))

3
```

```
In [67]: print(str(True),end = " ")
         int("4.5")

True
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[67], line 2
      1 print(str(True),end = " ")
----> 2 int("4.5")

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

```
In [75]: set = {1,3,6,5,4}
         set

Out[75]: {1, 3, 4, 5, 6}
```

```
In [70]: print(bool(0),bool(3.14),bool(-3),bool(1.0+1j))

False True True True
```

```
In [71]: s='ABC'
         type(s)

Out[71]: str
```

```
In [72]: x=10
         y="20"
         print(x+int(y))

30
```

```
In [77]: x={1,2,3}
         x.clear()
         print(x)
```

```
set()
```

In [79]:

```
x="hello"
y=x.upper()
y
```

Out[79]:

```
'HELLO'
```

In [80]:

```
x="hello"
y=x.replace('l','L',1)
y
```

Out[80]:

```
'heLlo'
```

In [81]:

```
x=(1,2,[3,4])
x[2][0]=5
x
```

Out[81]:

```
(1, 2, [5, 4])
```

In [82]:

```
type(x)
```

Out[82]:

```
tuple
```

In [83]:

```
x=[1,2,3]
y=x[:]
x[0]=4
y
```

Out[83]:

```
[1, 2, 3]
```

In [86]:

```
x={"apple","banana","cherry"}
y = x.pop()
y
```

Out[86]:

```
'apple'
```

In [87]:

```
#Area of rectangle
length = 45
width = 76
area = length * width
print(area)
```

```
3420
```

In [88]:

```
#Area of circle
pi =3.14
radius = 8.9
area = pi * radius * radius
print(area)
```

```
248.71940000000004
```

In [90]:

```
#String datatype
a = "Learning "
```

```
b = "is fun!"
c = a+b
print(c)
```

```
Learning is fun!
```

```
In [ ]:
```