Tuple

```
In [2]:  t1 = ()
```

```
In [3]:  t2 = (10,30,60) # tuple of integer numbers
```

```
In [4]:  t3 = (10.77,30.66,60.89) # tuple of float numbers
```

```
In [5]:  t4 = ('one','two','three') # tuple of strings
```

```
In [6]:  t5 = ('Asif',25,(50,100),(100,90)) # Nested tuples
```

```
In [7]:  t6 = (100,'Asif',17.765) # mixed data types
```

```
In [8]:  t7 = ('Asif',25,[50,100],[150,90],{'John','David'},(99,22,33))
```

```
In [9]:  len(t7)
```

```
Out[9]:  6
```

Tuple indexing

```
In [10]:  t2[0] # Retreive first element of the tuple
```

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

```
In [11]:  t4[0] # Retreive first element of the tuple
```

```
Out[11]:  'one'
```

```
In [12]:  t4[0][0]
```

```
Out[12]:  'o'
```

```
In [13]:  t4[-1] # Last element of the tuple
```

```
Out[13]:  'three'
```

```
In [14]:  t5[-1] # Last element of the tuple
```

```
Out[14]:  (100, 90)
```

Tuple Slicing

```
In [15]:  mytuple = ('one','two','three','four','five','six','seven','eight')
```

```
In [16]:  mytuple[0:3]
```

```
Out[16]:  ('one', 'two', 'three')
```

```
In [17]: mytuple[2:5]
```

```
Out[17]: ('three', 'four', 'five')
```

```
In [20]: mytuple[:3] # return first three items
```

```
Out[20]: ('one', 'two', 'three')
```

```
In [21]: mytuple[:2]  # return first two items
```

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

```
In [23]: mytuple[-3:] # return last three items
```

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

```
In [24]: mytuple[-2:] # return last two items
```

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

```
In [25]: mytuple[-1] # return last iten of the tuple
```

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

```
In [26]: mytuple[:] # return whole tuple
```

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

Remove & Change Items

```
In [28]: mytuple
```

```
Out[28]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [29]: for i in mytuple:
             print(i)
```

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

```
In [30]: for i in enumerate(mytuple):
             print(i)
```

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

Tuple Membership

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

```
Out[32]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [33]: 'one' in mytuple # check if 'one' exist in the list
```

```
Out[33]: True
```

```
In [35]: 'ten' in mytuple # check if 'ten' exist in the list
```

```
Out[35]: False
```

```
In [37]: if 'three' in mytuple: # check if three exist in the list
             print('Three is present in the tuple')
         else:
             print('Three is not present in the tuple')
```

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

```
In [38]: if 'eleven' in mytuple: # check if eleven exist in the list
             print('eleven is present in the tuple')
         else:
             print('eleven is not present in the tuple')
```

```
eleven is not present in the tuple
```

Index position

```
In [43]: mytuple
```

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

```
In [44]: mytuple.index('one')
```

```
Out[44]: 0
```

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

```
Out[45]: 4
```

```
In [46]: mytuple1 = ('one','two','three','four','one','one','two','three')
```

```
In [47]: mytuple1
```

```
Out[47]: ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

```
In [48]: mytuple1.index('one')

Out[48]: 0
```

Sorting

```
In [49]: mytuple2 = (43,67,99,12,6,90,67)

In [50]: mytuple2

Out[50]: (43, 67, 99, 12, 6, 90, 67)

In [57]: sorted(mytuple2)

Out[57]: [6, 12, 43, 67, 67, 90, 99]

In [59]: sorted(mytuple2, reverse=True) # sort in descending order

Out[59]: [99, 90, 67, 67, 43, 12, 6]
```

Bitwise Number System

```
In [60]: 25

Out[60]: 25

In [61]: bin(25)

Out[61]: '0b11001'

In [62]: int(0b11001)

Out[62]: 25

In [63]: bin(35)

Out[63]: '0b100011'

In [64]: oct(25)

Out[64]: '0o31'

In [66]: int(0o31)

Out[66]: 25

In [67]: bin(7)

Out[67]: '0b111'

In [68]: hex(7)
```

```
Out[68]: '0x7'
```

```
In [69]: hex(256)
```

```
Out[69]: '0x100'
```

```
In [ ]:
```