

# Experiment No-05: Tuple, Set, Comparison Operator and Logical Operator

**Name: Ravina Maruti Mane**

**Roll No: 27**

**Date:**

**Check Date:**

## Tuple

```
In [1]: t1=(12,13,14)
```

```
In [2]: t1
```

```
Out[2]: (12, 13, 14)
```

```
In [3]: type(t1)
```

```
Out[3]: tuple
```

```
In [5]: t1[2]
```

```
Out[5]: 14
```

```
In [6]: t2=('Ravina','Mane')
```

```
In [7]: t2[0]
```

```
Out[7]: 'Ravina'
```

```
In [8]: t2[0][2]
```

```
Out[8]: 'v'
```

```
In [9]: t2[0][2:3]
```

```
Out[9]: 'v'
```

```
In [10]: t2[0][::-1]
```

```
Out[10]: 'anivaR'
```

```
In [11]: t2.count('R')
```

```
Out[11]: 0
```

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

```
Out[12]: ('Ravina', 'Mane')
```

```
In [13]: t2[1]
```

```
Out[13]: 'Mane'
```

```
In [14]: t2[1]='Reva'
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[14], line 1  
----> 1 t2[1]='Reva'  
  
TypeError: 'tuple' object does not support item assignment
```

```
In [15]: t2.index('Maruti')
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[15], line 1  
----> 1 t2.index('Maruti')  
  
ValueError: tuple.index(x): x not in tuple
```

```
In [16]: t3=('Shree',100,300,600,'Swami')
```

```
In [17]: type(t3)
```

```
Out[17]: tuple
```

```
In [18]: t3.index(100)
```

```
Out[18]: 1
```

```
In [19]: t3.index('Swami')
```

```
Out[19]: 4
```

```
In [20]: t3.index('Swami1')
```

-----  
**ValueError**

Traceback (most recent call last)

Cell In[20], line 1

----> 1 t3.index('Swami1')

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

## Set

```
In [21]: s1={1,2,3,4,5}
```

```
In [22]: type(s1)
```

```
Out[22]: set
```

```
In [23]: s1
```

```
Out[23]: {1, 2, 3, 4, 5}
```

```
In [24]: s2={12,13,14,15,16}
```

```
In [25]: s2
```

```
Out[25]: {12, 13, 14, 15, 16}
```

```
In [26]: s2[1]
```

-----  
**TypeError**

Traceback (most recent call last)

Cell In[26], line 1

----> 1 s2[1]

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

```
In [27]: s3={1,1,1,2,2,2,3,3,3,}
```

```
In [28]: s3
```

```
Out[28]: {1, 2, 3}
```

```
In [29]: s3.update('4')
```

```
In [30]: s3
```

```
Out[30]: {1, 2, 3, '4'}
```

```
In [31]: s3.remove('4')
```

```
In [32]: s3
```

```
Out[32]: {1, 2, 3}
```

```
In [33]: s2
```

```
Out[33]: {12, 13, 14, 15, 16}
```

```
In [34]: s2.pop()
```

```
Out[34]: 16
```

```
In [35]: s4=s2.union(s3)
```

```
In [36]: s4
```

```
Out[36]: {1, 2, 3, 12, 13, 14, 15}
```

```
In [37]: s5=s3.union(s2)
```

```
In [38]: s5
```

```
Out[38]: {1, 2, 3, 12, 13, 14, 15}
```

```
In [39]: s5.update('Hello')
```

```
In [41]: s5
```

```
Out[41]: {1, 12, 13, 14, 15, 2, 3, 'H', 'e', 'l', 'o'}
```

## Comparison Operator And Logical Operator

### Comparison Operator

```
In [42]: 5>9
```

```
Out[42]: False
```

```
In [43]: 9>5
```

```
Out[43]: True
```

```
In [44]: 5>=7
```

```
Out[44]: False
```

```
In [45]: 5<=3
```

```
Out[45]: False
```

```
In [46]: 5<=5
```

```
Out[46]: True
```

```
In [47]: 5==5
```

```
Out[47]: True
```

```
In [48]: 5==6
```

```
Out[48]: False
```

```
In [51]: 5!=5
```

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

```
In [52]: 5!=6
```

```
Out[52]: True
```

```
In [53]: 'Ravina'=='Ravina'
```

```
Out[53]: True
```

```
In [54]: 'Ravina'=='ravina'
```

```
Out[54]: False
```

```
In [55]: int('Ravina'=='ravina')
```

```
Out[55]: 0
```

```
In [56]: 5>3
```

```
Out[56]: True
```

```
In [57]: int(5>3)
```

```
Out[57]: 1
```

```
In [58]: 4=5
```

```
Cell In[58], line 1
```

```
4=5
```

```
^
```

```
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?
```

### Logical Operator

```
In [59]: 4==4 and 5==5
```

```
Out[59]: True
```

```
In [60]: 4==3 and 5==5
```

```
Out[60]: False
```

```
In [61]: 4==3 and 5==4
```

```
Out[61]: False
```

```
In [62]: 3>2 and 2>3
```

```
Out[62]: False
```

```
In [63]: 3>2 or 2>3
```

```
Out[63]: True
```

```
In [64]: 3>4 or 2>3
```

```
Out[64]: False
```

```
In [65]: 3>2 or 2>3 and 2==2
```

```
Out[65]: True
```

```
In [66]: 3>2 or 2>3 and 2==3
```

```
Out[66]: True
```

```
In [67]: 3>2 and 2>3 and 2==3
```

```
Out[67]: False
```

```
In [68]: 3>4 and 2>3 and 2==3
```

```
Out[68]: False
```