

```
l = [1,2,3]
l
```

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

```
l.sort()
```

```
l
```

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

```
l1 = [20, 9, 3, 100]
l1
```

```
[20, 9, 3, 100]
```

```
l1.sort()
```

```
l1
```

```
[3, 9, 20, 100]
```

```
l2 = ['a', 3, 3.4, 1+2j]
l2
```

```
['a', 3, 3.4, (1+2j)]
```

```
l2.sort()
```

```
-----
TypeError
Cell In[12], line 1
----> 1 l2.sort()
```

Traceback (most recent call last)

TypeError: '<' not supported between instances of 'int' and 'str'

```
l3 = ['z', 'm', 'a', 'd' ]
l3
```

```
['z', 'm', 'a', 'd']
```

```
l3.sort()
```

```
l3
```

```
['a', 'd', 'm', 'z']
```

```
l3.reverse()
```

```
l3
```

```
['z', 'm', 'd', 'a']
```

```
l3.reverse()
```

```
l3
```

```
['a', 'd', 'm', 'z']
```

```
l1
```

```
[3, 9, 20, 100]
```

```
l1.sort(reverse=True)
```

```
l1
```

```
[100, 20, 9, 3]
```

```
l
```

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

```
l.append('nit')
```

```
l
```

```
[1, 2, 3, 'nit']
```

```
l[3]
```

```
'nit'
```

```
print(l[3][0])
```

```
print(l[3][1])
```

```
n
```

```
i
```

```
l
```

```
[1, 2, 3, 'nit']
```

```
for i in l:  
    print(i)
```

```
1
```

```
2
```

```
3
```

```
nit
```

```
for i in enumerate(l):  
    print(i)
```

```
(0, 1)
```

```
(1, 2)
(2, 3)
(3, 'nit')
```

```
l3
```

```
['a', 'd', 'm', 'z']
```

```
for j in enumerate(l3):
    print(j)
```

```
(0, 'a')
(1, 'd')
(2, 'm')
(3, 'z')
```

```
l
```

```
[1, 2, 3, 'nit']
```

```
all(l)
```

```
True
```

```
any(l)
```

```
True
```

```
l.append(0)
l
```

```
[1, 2, 3, 'nit', 0]
```

```
all(l)
```

```
False
```

```
any(l)
```

```
True
```

we completed LIST datastructure

TUPLE

```
t = ()
t
```

```
()
```

```
type(t)
```

```
tuple
```

```
t1 = (10, 20, 30, 40, 10)
t1
```

```
(10, 20, 30, 40, 10)
```

```
t1.append()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[45], line 1
----> 1 t1.append()
```

```
AttributeError: 'tuple' object has no attribute 'append'
```

```
t1
```

```
(10, 20, 30, 40, 10)
```

```
t1.index(40)
```

```
3
```

```
t1.count(20)
```

```
1
```

```
t1
```

```
(10, 20, 30, 40, 10)
```

```
t2 = t1.copy()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[52], line 1
----> 1 t2 = t1.copy()
```

```
AttributeError: 'tuple' object has no attribute 'copy'
```

```
t1
```

```
(10, 20, 30, 40, 10)
```

```
t2 = t1 * 2
```

```
t2
```

```
(10, 20, 30, 40, 10, 10, 20, 30, 40, 10)
```

```
icici= (1234, 'pan cizps67896', 6798)
```

```
icici
```

```
(1234, 'pan cizps67896', 6798)
```

```
t1
```

```
(10, 20, 30, 40, 10)
```

```
t1[0]
```

```
10
```

```
t1[0] = 100
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[61], line 1  
----> 1 t1[0] = 100
```

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

```
l
```

```
[1, 2, 3, 'nit', 0]
```

```
l[-1] = 0.3
```

```
l
```

```
[1, 2, 3, 'nit', 0.3]
```

```
t2 = ([1,2,3], 34, 5.6)
```

```
t2
```

```
([1, 2, 3], 34, 5.6)
```

```
len(t2)
```

```
3
```

```
l7 = [(1,2,3), [1,2]]
```

```
l7
```

```
[(1, 2, 3), [1, 2]]
```

tuple we are completed

22nd

```
s = {}
```

```
type(s)
```

```
dict
```

```
s = set()
```

```
s
```

```
set()
```

```
type(s)
```

```
set
```

```
s1 = {90, 80, 3, 13, 47, 100, 1}
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100}
```

```
s2 = {4, 'nit', 2.3, True, 1+2j}
```

```
s2
```

```
{(1+2j), 2.3, 4, True, 'nit'}
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100}
```

```
s1.add(200)
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s1.add(100)
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s3 = s1.copy()
```

```
s3
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s1 == s3
```

```
True
```

```
len(s3)
```

```
8
```

```
s3
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s3.clear()
```

```
s3
```

```
set()
```

```
len(s3)
```

```
0
```

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s1[:]
```

```
-----  
TypeError  
Cell In[19], line 1  
----> 1 s1[:]
```

Traceback (most recent call last)

TypeError: 'set' object is not subscriptable

```
s1
```

```
{1, 3, 13, 47, 80, 90, 100, 200}
```

```
s1.pop()
```

```
80
```

```
s1
```

```
{1, 3, 13, 47, 90, 100, 200}
```

```
s1.pop()
```

```
s1
```

```
{3, 13, 47, 90, 100, 200}
```

```
s1
```

```
{3, 13, 47, 90, 100, 200}
```

```
s1.pop(1)
```

```
-----  
TypeError  
Cell In[25], line 1  
----> 1 s1.pop(1)
```

Traceback (most recent call last)

TypeError: set.pop() takes no arguments (1 given)

```
s1
```

```
{3, 13, 47, 90, 100, 200}
```

```
s1.remove(1000)
```

```
-----  
KeyError
```

```
Traceback (most recent call last)
```

```
Cell In[27], line 1
```

```
----> 1 s1.remove(1000)
```

```
KeyError: 1000
```

```
s1.discard(1000)
```

```
s1
```

```
{3, 13, 47, 90, 100, 200}
```

```
s1.remove(47)
```

```
s1
```

```
{3, 13, 90, 100, 200}
```

```
s1.discard(200)
```

```
s1
```

```
{3, 13, 90, 100}
```

```
# basic set function we are completed
```

```
for i in s1:
```

```
    print(i)
```

```
3
```

```
100
```

```
90
```

```
13
```

```
for i in enumerate(s1):
```

```
    print(i)
```

```
(0, 3)
```

```
(1, 100)
```

```
(2, 90)
```

```
(3, 13)
```

```
s1
```

```
{3, 13, 90, 100}
```

```
s1.update([1,2])
```



s1

{1, 2, 3, 13, 90, 100}

set operation

A = {1,2,3,4,5}

B = {4,5,6,7,8}

C = {8,9,10}

A | B

{1, 2, 3, 4, 5, 6, 7, 8}

A.union(B)

{1, 2, 3, 4, 5, 6, 7, 8}

B.union(C)

{4, 5, 6, 7, 8, 9, 10}

A.union(C)

{1, 2, 3, 4, 5, 8, 9, 10}

print(A)

print(B)

print(C)

{1, 2, 3, 4, 5}

{4, 5, 6, 7, 8}

{8, 9, 10}

A.intersection(B)

{4, 5}

B & C

{8}

print(A)

print(B)

print(C)

{1, 2, 3, 4, 5}

{4, 5, 6, 7, 8}

{8, 9, 10}

A - B

{1, 2, 3}

C.difference(B)

{9, 10}

print(A)

print(B)

print(C)

{1, 2, 3, 4, 5}

{4, 5, 6, 7, 8}

{8, 9, 10}

A.symmetric\_difference(B)

{1, 2, 3, 6, 7, 8}

A.difference(B)

{1, 2, 3}

B ^ C

{4, 5, 6, 7, 9, 10}

print(A)

print(B)

print(C)

{1, 2, 3, 4, 5}

{4, 5, 6, 7, 8}

{8, 9, 10}

len(A)

5

A.symmetric\_difference\_update(B)

A

{1, 2, 3, 6, 7, 8}

print(A)

print(B)

print(C)

{1, 2, 3, 6, 7, 8}

{4, 5, 6, 7, 8}

{8, 9, 10}

B.symmetric\_difference\_update(C)

B

{4, 5, 6, 7, 9, 10}

A.symmetric\_difference(B)

{1, 2, 3, 4, 5}

A1 = {1,2,3,4,5,6,7,8,9}

B1 = {3,4,5,6,7,8}

C1 = {10,20,30,40}

A1.issuperset(B1)

True

B1.issubset(A1)

True

C1.isdisjoint(A1)

True

C1.isdisjoint(B1)

True

A2 = {1,2,3,4,5,6,7,8,9}

B2 = {30,40,50,60,70,80}

C2 = {10,20,30,40}

A2.issuperset(B2)

False

B2.issuperset(A2)

False

C2.isdisjoint(A2)

True

A2.isdisjoint(B2)

True

set is completed

DICTIONARY

d = {}

```
type(d)
```

```
dict
```

```
mydict = {1:'one' , 2:'two' , 3:'three' , 4:'four'}  
mydict
```

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
mydict[1]
```

```
'one'
```

```
mydict['one']
```

```
-----  
KeyError  
Cell In[86], line 1  
----> 1 mydict['one']
```

Traceback (most recent call last)

```
KeyError: 'one'
```

```
mydict.get(3)
```

```
'three'
```

24th

```
range(5)
```

```
range(0, 5)
```

```
list(range(0,5))
```

```
[0, 1, 2, 3, 4]
```

```
range(10,20)
```

```
range(10, 20)
```

```
list(range(10,20))
```

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
range(10,20,3)
```

```
range(10, 20, 3)
```

```
list(range(10,20,5))
```

```
[10, 15]
```

```
range(0,10,3,2)
```

-----  
TypeError Traceback (most recent call last)

Cell In[9], line 1

----> 1 range(0,10,3,2)

TypeError: range expected at most 3 arguments, got 4

r = range(0,11,5)

r

range(0, 11, 5)

for i in r:  
 print(i)

0

5

10

list(r)

[0, 5, 10]

advanced slicing

l = ['a', 'b', 'c', 1, 2.3, 45, True, 1+2j]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[:]

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[3:]

[1, 2.3, 45, True, (1+2j)]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[-5:]

[1, 2.3, 45, True, (1+2j)]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[5]

45

l[-5]

1

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[2:9]

['c', 1, 2.3, 45, True, (1+2j)]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[2:7]

['c', 1, 2.3, 45, True]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[2:7:2]

['c', 2.3, True]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[0:7:5]

['a', 45]

l

['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]

l[-5:3]

[]

l

`['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]`

`l[-5:-3]`

`[1, 2.3]`

`l`

`['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]`

`s = ['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`  
`s`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`

`len(s)`

`7`

`s`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`

`s[-7:-1]`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara']`

`s`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`

`s[::-1]`

`['city', 'canara', 'union', 'central', 'hdfc', 'icic', 'sbi']`

`s`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`

`s[::-2]`

`['city', 'union', 'hdfc', 'sbi']`

`s`

`['sbi', 'icic', 'hdfc', 'central', 'union', 'canara', 'city']`

`s[::-5]`

`['city', 'icic']`