7th march

SET

In [3]:
```
s={}
s
```
Out[3]:
```
{}
```

In [5]:
```
type(s) #set & dict both define with {}
```
Out[5]:
```
dict
```

In [7]:
```
s1=set()
type(s1)
```
Out[7]:
```
set
```

In [9]:
```
s1
```
Out[9]:
```
set()
```

In [11]:
```
s2={20,100,3,45}
s2
```
Out[11]:
```
{3, 20, 45, 100}
```

In [13]:
```
s3={'z','l','c','e','f'}
s3
```
Out[13]:
```
{'c', 'e', 'f', 'l', 'z'}
```

In [15]:
```
s4={1,2.3,'nit',1+2j,[1,2,3],(4,5,6),True}
s4
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 s4={1,2.3,'nit',1+2j,[1,2,3],(4,5,6),True}
      2 s4

TypeError: unhashable type: 'list'
```

In [55]:
```
s5={2,3.4,'nit',1+2j,False}
s5
```
Out[55]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [57]:
```
print(s1)
print(s2)
print(s3)
print(s5)
```

```
set()
{3, 100, 200, 45, 20, 30}
{'e', 'c', 'l', 'z', 'f'}
{False, 2, 3.4, 'nit', (1+2j)}
```

In [59]:
```
s2.add(30)
s2
```

Out[59]:
```
{3, 20, 30, 45, 100, 200}
```

In [61]:
```
s2.add(200)
s2
```

Out[61]:
```
{3, 20, 30, 45, 100, 200}
```

In [63]:
```
s2
```

Out[63]:
```
{3, 20, 30, 45, 100, 200}
```

In [65]:
```
s2[1:5]
s2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[65], line 1
----> 1 s2[1:5]
      2 s2

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

In [67]:
```
s5
```

Out[67]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [69]:
```
s4 =s5.copy()
s4
```

Out[69]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [71]:
```
s4
```

Out[71]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [73]:
```
s4.add(2) # duplicate is not allowed
s4
```

Out[73]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [75]:
```
s5
```

Out[75]:
```
{(1+2j), 2, 3.4, False, 'nit'}
```

In [77]:
```
s5.clear()
s5
```

Out[77]:
```
set()
```

```
In [81]: del s5
```

```
In [83]: s4
```

```
Out[83]: {(1+2j), 2, 3.4, False, 'nit'}
```

```
In [87]: s4.remove((1+2j))
```

```
In [89]: s4
```

```
Out[89]: {2, 3.4, False, 'nit'}
```

```
In [91]: s3
```

```
Out[91]: {'c', 'e', 'f', 'l', 'z'}
```

```
In [95]: s3.discard('m')#discard  never give error
```

```
In [97]: s3.remove('m')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[97], line 1
----> 1 s3.remove('m')

KeyError: 'm'
```

```
In [99]: s3.discard('f')
         s3
```

```
Out[99]: {'c', 'e', 'l', 'z'}
```

```
In [101… s3.pop()
```

```
Out[101… 'e'
```

```
In [103… s3
```

```
Out[103… {'c', 'l', 'z'}
```

```
In [105… s2
```

```
Out[105… {3, 20, 30, 45, 100, 200}
```

```
In [107… s2.pop(3)# indexing is not allowed
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[107], line 1
----> 1 s2.pop(3)

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

```
In [109… s2.pop()
```

```
Out[109… 3
```

In [111…

```
for i in s2:
    print(i)
```

```
100
200
45
20
30
```

In [ ]:

```
for
```

In [ ]:

In [113…

```
s2
```

Out[113…

```
{20, 30, 45, 100, 200}
```

In [115…

```
5 in s2
```

Out[115…

```
False
```

In [117…

```
45 in s2
```

Out[117…

```
True
```

In [119…

```
s2
```

Out[119…

```
{20, 30, 45, 100, 200}
```

In [125…

```
s3
```

Out[125…

```
{'c', 'l', 'z'}
```

In [121…

```
s2.update(s3)
```

In [123…

```
s2
```

Out[123…

```
{100, 20, 200, 30, 45, 'c', 'l', 'z'}
```

set operation

In [5]:

```
s6 ={1,2,3,4,5}
s7 ={4,5,6,7,8}
s8 ={8,9,10}
```

UNION

In [8]:

```
s6.union(s7)
```

Out[8]:

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

In [10]:

```
s6.union(s7,s8)
```

```
Out[10]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [12]: s6|s7
```

```
Out[12]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [14]: s6|s7|s8
```

```
Out[14]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [16]: print(s6)
         print(s7)
         print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

INTERSECTION

```
In [19]: s6.intersection(s7)
```

```
Out[19]: {4, 5}
```

```
In [21]: s6.intersection(s8)
```

```
Out[21]: set()
```

```
In [23]: s7.intersection(s8)
```

```
Out[23]: {8}
```

```
In [25]: s6&s7
```

```
Out[25]: {4, 5}
```

```
In [27]: print(s6)
         print(s7)
         print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

DIFFERENCE

```
In [30]: s6.difference(s7)
```

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

```
In [32]: s6 -s7
```

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

In [34]:

```
s7 -s8
```

Out[34]:

```
{4, 5, 6, 7}
```

In [36]:

```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [38]:

```
s8 -s7
```

Out[38]:

```
{9, 10}
```

In [40]:

```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

SYMMETRIC_DIFFERENCE

In [43]:

```
s6.symmetric_difference(s7)
```

Out[43]:

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

In [45]:

```
s7.symmetric_difference(s8)
```

Out[45]:

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

In [48]:

```
s6.symmetric_difference(s8)
```

Out[48]:

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

In [ ]: