Set Creation

In [4]:
```
myset = {1,2,3,4,5} #set of elements
myset
```

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

In [5]:
```
len(myset) #length of set
```

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

In [7]:
```
my_set = {1,1,2,3,4,5,5} #duplicates are not allowed in set
my_set
```

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

In [9]:
```
set1 = {2.3,3.2,4.5} #set of float numbers
set1
```

Out[9]:
```
{2.3, 3.2, 4.5}
```

In [10]:
```
set2 = {'ab','cd','ef'} #set of strings
set2
```

Out[10]:
```
{'ab', 'cd', 'ef'}
```

In [13]:
```
set3 = {1,2,"heaven",(3,4,5,)} #different data types
set3
```

Out[13]:
```
{(3, 4, 5), 1, 2, 'heaven'}
```

In [14]:
```
set4={1,3,'c',[1,2]} # it doesnt allow mutable list
set4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 set4={1,3,'c',[1,2]} # it doesnt allow mutable list
      2 set4

TypeError: unhashable type: 'list'
```

In [17]:
```
s = set() # creating an empty set
print(type(s))
```

```
<class 'set'>
```

In [19]:
```
s1 = set(('one','two','three'))
s1
```

Out[19]:

```
{'one', 'three', 'two'}
```

Loop through a Set

In [20]:

```
myset={'one','two','three','four','five'}
for i in myset:
    print(i)
```

```
one
two
three
four
five
```

In [21]:

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

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
```

Set Membership

In [22]:

```
myset
```

Out[22]:

```
{'five', 'four', 'one', 'three', 'two'}
```

In [23]:

```
'one' in myset
```

Out[23]:

```
True
```

In [24]:

```
'six' in myset
```

Out[24]:

```
False
```

In [25]:

```
if 'three' in myset:
    print('Three is present in the set')
else:
    print('Three is not present in the set')
```

```
Three is present in the set
```

In [26]:

```
if 'six' in myset:
    print('six is present in the set')
else:
    print('six is not present')
```

```
six is not present
```

Add & Remove Items

```
In [28]: myset
```

```
Out[28]: {'five', 'four', 'one', 'three', 'two'}
```

```
In [29]: myset.add('seven')
```

```
In [30]: myset
```

```
Out[30]: {'five', 'four', 'one', 'seven', 'three', 'two'}
```

```
In [31]: myset.update(['seven','eight','nine'])
         myset
```

```
Out[31]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'three', 'two'}
```

```
In [32]: myset.remove('three')
         myset
```

```
Out[32]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'two'}
```

```
In [34]: myset.discard('ten')
```

```
In [35]: myset.discard('seven')
```

```
In [36]: myset
```

```
Out[36]: {'eight', 'five', 'four', 'nine', 'one', 'two'}
```

```
In [37]: myset.clear()
         myset
```

```
Out[37]: set()
```

```
In [38]: del myset
         myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[38], line 2
      1 del myset
----> 2 myset

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

Copy Set

```
In [40]: myset = {1,2,3,4,5,6}
         myset
```

```
Out[40]: {1, 2, 3, 4, 5, 6}
```

```
In [41]: myset1 = myset
         myset1
```

```
Out[41]: {1, 2, 3, 4, 5, 6}
```

```
In [42]: id(myset),id(myset1)
```

```
Out[42]: (2119020340480, 2119020340480)
```

```
In [43]: my_set=myset.copy()
         my_set
```

```
Out[43]: {1, 2, 3, 4, 5, 6}
```

```
In [44]: id(my_set)
```

```
Out[44]: 2119020342496
```

```
In [45]: myset.add(7)
         myset
```

```
Out[45]: {1, 2, 3, 4, 5, 6, 7}
```

```
In [46]: myset1
```

```
Out[46]: {1, 2, 3, 4, 5, 6, 7}
```

```
In [47]: my_set
```

```
Out[47]: {1, 2, 3, 4, 5, 6}
```

Set Operation

union

```
In [48]: A={1,2,3,4,5}
         B={4,5,6,7,8}
         C={8,9,10}
```

```
In [49]: A | B #all elements
```

```
Out[49]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [50]: A.union(B)
```

```
Out[50]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [51]: A.union(B,C)
```

```
Out[51]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [52]: A.update(B,C) #update function will update
```

```
In [53]: A
```

```
Out[53]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Intersection

```
In [54]: A = {1,2,3,4}
         B = {3,4,6,7}
```

```
In [55]:  A & B
```

```
Out[55]: {3, 4}
```

```
In [58]: A.intersection(B) #intersection gives common elements
```

```
Out[58]: {3, 4}
```

```
In [57]: A.intersection_update(B)
         A
```

```
Out[57]: {3, 4}
```

Difference

```
In [60]: A = {1,2,3,4,5}
         B = {4,5,6,7,8}
```

```
In [61]: A - B
```

```
Out[61]: {1, 2, 3}
```

```
In [62]: A.difference(B)
```

```
Out[62]: {1, 2, 3}
```

```
In [64]: B.difference(A)
```

```
Out[64]: {6, 7, 8}
```

In [65]:
```
B.difference_update(A)
B
```
Out[65]:
```
{6, 7, 8}
```

Symmetric Difference

In [67]:
```
A={1,2,3,4,5}
B={4,5,6,7,8}
```

In [68]:
```
A.symmetric_difference(B) #set of elements in A and B but not in both)
```
Out[68]:
```
{1, 2, 3, 6, 7, 8}
```

Subset,Superset,Disjoint

In [69]:
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

In [70]:
```
B.issubset(A)
```
Out[70]:
```
True
```

In [71]:
```
A.issuperset(B)
```
Out[71]:
```
True
```

In [72]:
```
C.isdisjoint(A)
```
Out[72]:
```
True
```

In [73]:
```
B.isdisjoint(A)
```
Out[73]:
```
False
```

some built in function

In [74]:
```
A
```
Out[74]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [75]:
```
sum(A)
```
Out[75]:
```
45
```

In [76]:

```
max(A)
```

Out[76]:

```
9
```

In [77]:

```
min(A)
```

Out[77]:

```
1
```

In [78]:

```
len(A)
```

Out[78]:

```
9
```

In [79]:

```
list(enumerate(A))
```

Out[79]:

```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```

In [80]:

```
D = sorted(A,reverse = True)
D
```

Out[80]:

```
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [81]:

```
sorted(D)
```

Out[81]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```