Introduction to Sets

1) Unordered and Unindexed collection of items

2) Set elements are unique and duplicate elements are not allowed.

3) Set elements are immutable (i.e. cannot be changed).

4) Set itself is mutable. We can add or remove items from it.

Set Creation

In [11]:
```
mset = {1,2,3,4,5}  # Set of numbers
mset
```

Out[11]:
```
{1, 2, 3, 4, 5}
```

In [13]:
```
len(mset) #Length of the set
```

Out[13]:
```
5
```

In [17]:
```
#Checking duplicate elements if allowed
m_set = {1,1,2,2,3,4,5,5}
m_set                    # Duplicate elements not allowed
```

Out[17]:
```
{1, 2, 3, 4, 5}
```

In [23]:
```
m_set1 = {2.3,52.3,45.9,4.5,3.09}
m_set1      #Set of float numbers
```

Out[23]:
```
{2.3, 3.09, 4.5, 45.9, 52.3}
```

In [25]:
```
m_set2 = {'Sun','Moon','Planet'} #Set of Strings
m_set2
```

Out[25]: {'Moon', 'Planet', 'Sun'}

In [27]:
```
m_set3 = {9,34,"Sun",(33,53,67)}  #Mixed datatypes
m_set3
```

Out[27]: {(33, 53, 67), 34, 9, 'Sun'}

In [29]:
```
m_set4 = {4,"Moon",54.3,[1,5,6]} #Set does allow mutable items like list
m_set4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 m_set4 = {4,"Moon",54.3,[1,5,6]} #Set does allow mutable items like list
      2 m_set4

TypeError: unhashable type: 'list'
```

In [37]:
```
m_set5 = set()
print(type(m_set5))
```

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

In [39]:
```
my_set1 = set(('abc','def','ghi','jkl'))
my_set1
```

Out[39]: {'abc', 'def', 'ghi', 'jkl'}

Looping through a set

In [46]:
```
myset = {'one','two','three','four','five','six','seven'}
for i in myset:
    print(i)
```

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

In [50]:
```
for i in enumerate(myset):
    print(i)
```

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

Set Membership

In [53]:
```
myset
```

Out[53]:
```
{'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [55]:
```
'four' in myset
```

Out[55]:
```
True
```

In [57]:
```
"Nine" in myset
```

Out[57]:
```
False
```

In [59]:
```
if 'zero' in myset:
    print("Zero is present in the set")
else:
    print("Zero is not present in the set")
```

```
Zero is not present in the set
```

In [61]:
```
if 'seven' in myset:
    print("Seven is present in the set")
else:
    print("Seven is not present in the set")
```

```
Seven is present in the set
```

Add and Remove Items

In [64]:
```
myset
```

Out[64]:
```
{'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [66]:
```
myset.add('nine')
myset
```

Out[66]:
```
{'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [68]:
```
myset.update(['eleven','twelve','thirteen'])
myset
```

Out[68]:
```
{'eleven',
 'five',
 'four',
 'nine',
 'one',
 'seven',
 'six',
 'thirteen',
 'three',
 'twelve',
 'two'}
```

In [70]:
```
myset.remove('two')
myset
```

Out[70]:
```
{'eleven',
 'five',
 'four',
 'nine',
 'one',
 'seven',
 'six',
 'thirteen',
 'three',
 'twelve'}
```

In [72]:
```
myset.remove('fourteen')
myset           #while using remove - if element not found it gives error
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[72], line 1
----> 1 myset.remove('fourteen')
      2 myset

KeyError: 'fourteen'
```

In [74]:
```
myset.discard('three')
myset
```

Out[74]:
```
{'eleven', 'five', 'four', 'nine', 'one', 'seven', 'six', 'thirteen', 'twelve'}
```

In [76]:
```
myset.discard('fourteen')
myset            # if element not found it does not give error as remove() does
```

Out[76]:
```
{'eleven', 'five', 'four', 'nine', 'one', 'seven', 'six', 'thirteen', 'twelve'}
```

In [78]:
```
myset.clear() #deletes all the items in a set
myset
```

Out[78]:
```
set()
```

In [86]:
```
del myset
myset       #deletes the set object
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[86], line 1
----> 1 del myset
      2 myset

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

Copy Set

In [100…
```
ms = {'one','two','three','four','five','six','seven','thirteen','eighteen'}
```

In [102…
```
ms
```

Out[102…
```
{'eighteen', 'five', 'four', 'one', 'seven', 'six', 'thirteen', 'three', 'two'}
```

In [106…
```
myset1 = ms #Create a new reference "myset1"
myset1
```

Out[106…
```
{'eighteen', 'five', 'four', 'one', 'seven', 'six', 'thirteen', 'three', 'two'}
```

In [110…
```
id(myset1), id(ms) #The address of both myset1 and ms will be the same
```

Out[110…
```
(1345976767584, 1345976767584)
```

In [117…
```
my_set = myset1.copy()
my_set
```

Out[117…
```
{'eighteen', 'five', 'four', 'one', 'seven', 'six', 'thirteen', 'three', 'two'}
```

In [123…
```
myset1
```

Out[123…
```
{'eighteen', 'five', 'four', 'one', 'seven', 'six', 'thirteen', 'three', 'two'}
```

In [119…
```
id(my_set)
```

Out[119…
```
1345976769376
```

In [127…
```
my_set.add('twenty')
my_set
```

Out[127…
```
{'eighteen',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'thirteen',
 'three',
 'twenty',
 'two'}
```

In [129…
```
myset1
```

Out[129…
```
{'eighteen', 'five', 'four', 'one', 'seven', 'six', 'thirteen', 'three', 'two'}
```

Set Operation

In [132…
```
#Union
a = {1,2,3,4,5}
b = {4,5,6,7,8}
c = {8,9,10}
```

In [136…
```
a | b #Union of all elements, no duplicates
```

Out[136…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [138…
```
a.union(b)
```

Out[138…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [140…
```
a.union(b,c) #union of a,b,c
```

Out[140…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [142…
```
# Update() method
# Updates the set calling the update() method with union of a,b and c
a.update(b,c) #set a will be updated with the union of a,b,c
a
```

Out[142…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [151…
```
m = {2,3,4}
n = {5,6}
o = {3,5,6,7,9}
o.update(m,n)
o
```

Out[151…
```
{2, 3, 4, 5, 6, 7, 9}
```

In [161…
```
#Intersection
a = {1,2,3,7,8}
b = {4,5,6,7,8}
a.intersection(b)
```

Out[161…
```
{7, 8}
```

In [165…
```
a & b # Intersection
```

Out[165…
```
{7, 8}
```

In [169…
```
a.intersection_update(b) #updates the set calling intersection_update method with t
```

```
a
```

Out[169… {7, 8}

In [171…
```
b.intersection_update(a)
b
```

Out[171… {7, 8}

In [173…
```
#Difference
a = {1,2,3,4,5}
b = {4,5,6,7,8}
```

In [175…
```
a - b #set of elements that are only in a but not in b
```

Out[175… {1, 2, 3}

In [183…
```
a.difference(b)
```

Out[183… {1, 2, 3}

In [185…
```
b - a
```

Out[185… {6, 7, 8}

In [187…
```
b.difference(a)
```

Out[187… {6, 7, 8}

In [197…
```
b.difference_update(a)
b
```

Out[197… {4, 5, 6, 7, 8}

Symmetric Difference

In [202…
```
a = {1,2,3,4,5}
b = {4,5,6,7,8}
```

In [205…
```
a.symmetric_difference(b)
```

Out[205… {1, 2, 3, 6, 7, 8}

In [207…
```
a ^ b
```

Out[207… {1, 2, 3, 6, 7, 8}

In [209…
```
#symmetric_difference_update
a.symmetric_difference_update(b)
a
```

Out[209… {1, 2, 3, 6, 7, 8}

In [215…
```
a
```

Out[215… {1, 2, 3, 6, 7, 8}

In [217…
```
b
```

Out[217… {4, 5, 6, 7, 8}

In [219…
```
b.symmetric_difference_update(a)
b
```

Out[219… {1, 2, 3, 4, 5}

Subset, Superset and Disjoint

In [222…
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

In [224…
```
B.issubset(A)
```

Out[224… True

In [226…
```
A.issuperset(B)
```

Out[226… True

In [228…
```
A.issubset(B)
```

Out[228… False

In [232…
```
C.isdisjoint(A) #Two sets are said to be disjoint sets if they have no common eleme
```

Out[232… True

In [234…
```
B.isdisjoint(C)
```

Out[234… True

In [236…
```
A.isdisjoint(B)
```

Out[236… False

Other built in functions

In [239…
```
a
```

Out[239… {1, 2, 3, 6, 7, 8}

In [241…
```
sum(a)
```

Out[241… 27

In [243…
```
min(a)
```

Out[243… 1

In [245…
```
max(a)
```

Out[245… 8

In [249…
```
len(a)
```

Out[249… 6

In [251…
```
list(enumerate(a))
```

Out[251… [(0, 1), (1, 2), (2, 3), (3, 6), (4, 7), (5, 8)]

In [255…
```
d = sorted(a)
d
```

Out[255… [1, 2, 3, 6, 7, 8]

In [259…
```
d = sorted(a, reverse = True)
d
```

Out[259… [8, 7, 6, 3, 2, 1]

In [261…
```
sorted(d)
```

Out[261… [1, 2, 3, 6, 7, 8]