Sets

Set Creation

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

In [2]:
```
s1
```

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

In [3]:
```
len(s1)
```

Out[3]:
```
5
```

In [4]:
```
s2 = {1,1,2,2,2,3,3,4,4,5,6,7,7} # Duplicate elements are not allowed
s2
```

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

In [7]:
```
set1 = {1.2,3.5,4.56,6.78}   # Set of float numbers
set1
```

Out[7]:
```
{1.2, 3.5, 4.56, 6.78}
```

In [9]:
```
set2 = {'Ramu','Ashif','Mary'} # Set of Strings
set2
```

Out[9]:
```
{'Ashif', 'Mary', 'Ramu'}
```

In [10]:
```
set3 = {10,20,30.48,"Holy",(10,20,30)} # Mixed datatypes
set3
```

Out[10]:
```
{(10, 20, 30), 10, 20, 30.48, 'Holy'}
```

In [11]:
```
set4 = {10,20,"Holy",[10,20,30]}  # Set doesn't allow mutable items like List
set4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 set4 = {10,20,"Holy",[10,20,30]}
      2 set4

TypeError: unhashable type: 'list'
```

In [12]:
```
set3
```

Out[12]:
```
{(10, 20, 30), 10, 20, 30.48, 'Holy'}
```

In [13]:
```
print(type(set3))
```

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

Loop Through a Set

In [14]:
```
myset = {'one','two','three','four','five','six','seven','eight'}

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

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

In [15]:
```
for i in enumerate(myset):
    print(i)
```

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

Set Membership

In [16]:
```
myset
```

Out[16]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [17]:
```
'five' in myset
```

Out[17]:
```
True
```

In [19]:
```
'four' in myset
```

Out[19]:
```
True
```

In [20]:
```
'ten' in myset
```

Out[20]:
```
False
```

In [22]:
```
if 'one' in myset:
    print("one is present in myset")
else:
    print("Not present")
```

```
one is present in myset
```

In [23]:
```
if 'ten' in myset:
    print("ten present in myset")
else:
    print("Ten is not present")
```

```
Ten is not present
```

Add & Remove Items

In [24]:
```
myset
```

Out[24]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [25]:
```
myset.add('NINE') # Add item to a set using add()
myset
```

Out[25]:
```
{'NINE', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [29]:
```
myset.update(['TEN','ELEVEN','TWELVE'])
myset
```

Out[29]:
```
{'E',
 'ELEVEN',
 'L',
 'N',
 'NINE',
 'T',
 'TEN',
 'TWELVE',
 'V',
 'W',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

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

Out[32]:
```
{'E',
 'ELEVEN',
 'L',
 'N',
 'T',
 'TEN',
 'TWELVE',
 'V',
 'W',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

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

Out[34]:
```
{'E',
 'ELEVEN',
 'L',
 'N',
 'T',
 'TWELVE',
 'V',
 'W',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [35]:
```
myset.clear()
myset
```

Out[35]:
```
set()
```

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[36], line 2
      1 del myset
----> 2 myset

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

Copy Set

In [37]:
```
myset1 = {'one','two','three','four','five','six','seven','eight'}
```

In [38]:
```
myset1
```

Out[38]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [39]:
```
myset2 = myset1
myset2
```

Out[39]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [40]:
```
id(myset2), id(myset1)
```

Out[40]:
```
(1792118168192, 1792118168192)
```

In [42]:
```
my_set = myset1.copy()
my_set
```

Out[42]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [43]:
```
id(my_set)
```

Out[43]: 1792118172896

In [44]:
```
myset1
```

Out[44]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [45]:
```
my_set
```

Out[45]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

Set Operation

Union

In [46]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [47]:
```
A | B       # Union of A and B (All Elements from both sets. NO DUPLICATES)
```

Out[47]: {1, 2, 3, 4, 5, 6, 7, 8}

In [53]:
```
A.union(B)
```

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

In [56]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
A.union(B,C)   # Union of A,B, and C
```

Out[56]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [58]:
```
A.update(B,C)
A
```

Out[58]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Intersection

In [59]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [60]:
```
A & B     # Intersection in A and B (common elements in both sets)
```

Out[60]: {4, 5}

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

Out[62]: {4, 5}

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

Out[64]: {4, 5}

Difference

In [65]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [66]:
```
A - B      # Set of elements that are only in A but not in B
```

Out[66]: {1, 2, 3}

In [67]:
```
A.difference(B)
```

Out[67]: {1, 2, 3}

In [68]:
```
B - A
```

Out[68]: {6, 7, 8}

In [69]:
```
B.difference(A)
```

Out[69]: {6, 7, 8}

In [71]:
```
B.difference_update(A)
B
```

Out[71]: {6, 7, 8}

Symmetric Difference

In [72]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [73]:
```
A ^ B     # Symmetric Difference (set of elements in A and B but not both)
```

Out[73]: {1, 2, 3, 6, 7, 8}

In [74]:
```
A.symmetric_difference(B)
```

Out[74]: {1, 2, 3, 6, 7, 8}

In [76]:
```
A.symmetric_difference_update(B)
A
```

Out[76]: {1, 2, 3, 4, 5}

Subset, Superset & Disjoint

In [77]:
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

In [78]:
```
B.issubset(A)
```

Out[78]: True

In [79]:
```
A.issuperset(B)
```

Out[79]: True

In [80]:
```
C.isdisjoint(A)
```

Out[80]: True

Other Built-in Functions

In [81]:
```
A
```

Out[81]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [82]:
```
sum(A)
```

Out[82]: 45

In [83]:
```
max(A)
```

Out[83]: 9

In [84]:
```
min(A)
```

Out[84]: 1

In [85]:
```
len(A)
```

Out[85]: 9

In [86]:
```
list(enumerate(A))
```

Out[86]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]

In [88]:
```
D = sorted(A, reverse=True)
D
```

Out[88]: [9, 8, 7, 6, 5, 4, 3, 2, 1]

In [89]:
```
sorted(D)
```

Out[89]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [90]:
```
A
```

Out[90]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: