Set Data Structure

Sets are used to store multiple items in a single variable. Set is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Tuple, and Dictionary, all with different qualities and usage. A set is a collection which is unordered, unchangeable*, and unindexed.

Set Creation

In [1]:
```
set1 = {1,2,3,4,5,6}#Set data structure of numbers
set1
```

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

In [2]:
```
len(set1)# Length of the set
```

Out[2]:
```
6
```

In [4]:
```
set2 = {1,1,2,2,3,3,4,4,4,5,5,5,6,6,7,7,7,7,8,9,10}
set2 # Duplicate entries are not allowed.
```

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

In [9]:
```
#Nested Set ,Case 1
set3 = {1,2,3,4,("Sneha","Singh"),[2.3,4.3,]}
set3 # We cannot nest the list inside the set
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7984\3577709750.py in <module>
      1 #Nested Set ,Case 1
----> 2 set3 = {1,2,3,4,("Sneha","Singh"),[2.3,4.3,]}
      3 set3 # We cannot nest the list inside the set

TypeError: unhashable type: 'list'
```

In [7]:
```
# Nested Set ,Case 2
set4 = {1,2,3,4,5,("Lovely","Singh")}
set4 # We can nest the tuple inside the set
```

Out[7]:
```
{('Lovely', 'Singh'), 1, 2, 3, 4, 5}
```

Set Membership

In [19]:
```
# dont know how to check
"Singh" in set4
```

Out[19]:
```
False
```

In [18]:
```
4 in set4
```

Out[18]:
```
True
```

Add and Remove Items

In [22]:
```
set5 = {"Apple","Orange","Strawberry","Mango"}
set5
```

Out[22]:
```
{'Apple', 'Mango', 'Orange', 'Strawberry'}
```

In [23]:
```
set5 .add("Peach")#Add items to set using add () method
set5
```

Out[23]:
```
{'Apple', 'Mango', 'Orange', 'Peach', 'Strawberry'}
```

In [46]:
```
# Add multiple item to a set using square brackets
set1.update(["Kiwi","Promegrenate","Grapes"])
set1
```

Out[46]:
```
{1, 2, 3, 4, 5, 6, 'Grapes', 'Kiwi', 'Promegrenate'}
```

In [49]:
```
set1.remove("Promegrenate")# remove the item using the remove()method
set1
```

Out[49]:
```
{1, 2, 3, 4, 5, 6, 'Grapes'}
```

In [50]:
```
set1.discard(4)# Remove the item using discard() method
set1
```

Out[50]:
```
{1, 2, 3, 5, 6, 'Grapes'}
```

In [52]:
```
set1.clear()#Delete all items in a set
set1
```

Out[52]:
```
set()
```

In [55]:
```
del set1 # delete the set object
set1 # already delted
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7984\3088987097.py in <module>
----> 1 del set1 # delete the set object
      2 set1 # already delted

NameError: name 'set1' is not defined
```

Copy Set

In [56]:
```
new_set = {1,2,3,4,5}
new_set
```

Out[56]:
```
{1, 2, 3, 4, 5}
```

In [57]:
```
new_set_1 = new_set # create a new refernce "new_set_1"
new_set_1
```

Out[57]:
```
{1, 2, 3, 4, 5}
```

In [58]:
```
#The address of both "new_set" and the "new_set_1"
id(new_set),id(new_set_1)
```

Out[58]: (2013977691296, 2013977691296)

In [59]:
```
newset2 = new_set_1.copy()# Create copy of list
newset2
```

Out[59]: {1, 2, 3, 4, 5}

Set Operation

UNION

In [60]:
```
#Union
A = {1,2,3,4,5,6}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [61]:
```
A|B # Union of A and B (all elements from both the sets.NO Duplicates)
```

Out[61]: {1, 2, 3, 4, 5, 6, 7, 8}

In [64]:
```
A .union (B)# Union of A and B
```

Out[64]: {1, 2, 3, 4, 5, 6, 7, 8}

In [65]:
```
A .union(B,C)# Union of A,B and C
```

Out[65]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Intersection

In [66]:
```
A & B #Intersection of A and B (Common items in both sets)
```

Out[66]: {4, 5, 6}

Difference

In [67]:
```
A-B # set of elements that are only in A but not in B
```

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

In [68]:
```
A .difference(B)# Difference of sets
```

Out[68]: {1, 2, 3}

Symmetric Difference

In [69]:
```
A^B # Symmetric difference (set of elements in A and B but not in both)
```

Out[69]: {1, 2, 3, 7, 8}

In [70]:
```
A . symmetric_difference(B)# Symmetric difference of sets
```

Out[70]: {1, 2, 3, 7, 8}

Subset ,Superset and Disjoint

In [77]:
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7}
C = {11,12,1,31,4,15,16}
```

In [73]:
```
B .issubset(A)
```

Out[73]:
```
True
```

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

Out[74]:
```
True
```

In [76]:
```
C .isdisjoint(A)
```

Out[76]:
```
False
```

Built in Functions

In [79]:
```
sum(A)
```

Out[79]:
```
45
```

In [80]:
```
max(A)
```

Out[80]:
```
9
```

In [81]:
```
min(A)
```

Out[81]:
```
1
```

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

Out[82]:
```
9
```

In [83]:
```
list(enumerate(A))
```

Out[83]:
```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```

In [84]:
```
D = sorted(A,reverse = True)
D
```

Out[84]:
```
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [85]:
```
sorted(D)
```

Out[85]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```