SET

Unordered and unidexed collections of items.

```
In [9]:  s={}
```

```
In [10]: type(s)
```

```
Out[10]: dict
```

```
In [11]: s={1,}
```

```
In [12]: type(s)
```

```
Out[12]: set
```

```
In [13]: S1={10,20,30,40,50}
```

```
In [14]: len(S1)
```

```
Out[14]: 5
```

```
In [15]: s2={10,20,20,30,40,50,60} #Duplicate are not allowed
         s2
```

```
Out[15]: {10, 20, 30, 40, 50, 60}
```

```
In [16]: s3={10.0,20.0,30.0,40.0,10.0,30.0,40.0} #set of float number
         s3
```

```
Out[16]: {10.0, 20.0, 30.0, 40.0}
```

```
In [17]: s4={'abc','def','xyz'} #set of string
         s4
```

```
Out[17]: {'abc', 'def', 'xyz'}
```

```
In [20]: s5={10,20.0,'abc'}
         s5
```

```
Out[20]: {10, 20.0, 'abc'}
```

Set functions

.add()

```
In [21]: my_set={100,200,300,400,500,600}
         my_set
```

```
Out[21]: {100, 200, 300, 400, 500, 600}
```

```
In [22]: my_set.add(700) #It is used to add a new element into a set
         my_set
```

```
Out[22]: {100, 200, 300, 400, 500, 600, 700}
```

```
In [24]: my_set.add(800)
         my_set
```

```
Out[24]: {100, 200, 300, 400, 500, 600, 700, 800}
```

.copy()

```
In [26]: my_set
```

```
Out[26]: {100, 200, 300, 400, 500, 600, 700, 800}
```

```
In [31]: my_set1=my_set.copy() #This function creates and return a duplicate of the set
         my_set1
```

```
Out[31]: {100, 200, 300, 400, 500, 600, 700, 800}
```

```
In [33]: myset2=my_set1.copy()
         myset2
```

```
Out[33]: {100, 200, 300, 400, 500, 600, 700, 800}
```

.clear()

```
In [34]: myset2.clear() # Remove all item from the set
         myset2
```

```
Out[34]: set()
```

```
In [36]: my_set1.clear()
         my_set1
```

```
Out[36]: set()
```

.update()

```
In [37]: my_set
```

```
Out[37]: {100, 200, 300, 400, 500, 600, 700, 800}
```

```
In [43]: my_set.update([150,250,350]) #adds multiple elemnt in to set
         my_set
```

```
Out[43]: {'0',
          '1',
          100,
          150,
          '2',
          200,
          250,
          '3',
          300,
          350,
          400,
          '5',
          500,
          600,
          700,
          800,
          'E',
          'H',
          'N',
          'O',
          'R',
          'T',
          'W'}
```

```
In [46]: my_set.update([180,280,890])
         my_set
```

```
Out[46]: {'0',
          '1',
          100,
          150,
          180,
          '2',
          200,
          250,
          280,
          '3',
          300,
          350,
          400,
          '5',
          500,
          600,
          700,
          800,
          890,
          'E',
          'H',
          'N',
          'O',
          'R',
          'T',
          'W'}
```

.discard()

```
In [60]: my_set.discard('5') #removes items from set and doesn't give error if the element i
         my_set
```

```
Out[60]: {100,
          150,
          180,
          200,
          250,
          280,
          300,
          350,
          400,
          500,
          600,
          700,
          800,
          890,
          'E',
          'H',
          'N',
          'O',
          'R',
          'T',
          'W'}
```

```
In [64]: my_set.discard('100000')
         my_set
```

```
Out[64]: {100,
          150,
          180,
          200,
          250,
          280,
          300,
          350,
          400,
          500,
          600,
          700,
          800,
          890,
          'H',
          'N',
          'O',
          'R',
          'T',
          'W'}
```

.pop()

```
In [65]: my_set
```

```
Out[65]: {100,
          150,
          180,
          200,
          250,
          280,
          300,
          350,
          400,
          500,
          600,
          700,
          800,
          890,
          'H',
          'N',
          'O',
          'R',
          'T',
          'W'}
```

```
In [66]: my_set.pop() # removes a random item from the set
         my_set
```

Out[66]:

```
{100,
 150,
 180,
 250,
 280,
 300,
 350,
 400,
 500,
 600,
 700,
 800,
 890,
 'H',
 'N',
 'O',
 'R',
 'T',
 'W'}
```

In [67]:

```
my_set.pop()
my_set
```

Out[67]:

```
{100,
 150,
 180,
 250,
 280,
 300,
 350,
 400,
 500,
 600,
 700,
 800,
 890,
 'H',
 'N',
 'O',
 'R',
 'W'}
```

SET OPERATIONS

UNION(|)

In [68]:

```
x={1,2,3,4,5,6}
y={5,6,7,8,9}
z={8,9,10}
```

In [69]:

```
x.union(y) #combine sets without duplicates
```

```
Out[69]: {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In [70]: x.union(z)
```

```
Out[70]: {1, 2, 3, 4, 5, 6, 8, 9, 10}
```

```
In [71]: y|z
```

```
Out[71]: {5, 6, 7, 8, 9, 10}
```

```
In [72]: z|y|x
```

```
Out[72]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

INTERSECTION(&)

```
In [74]: print(x)
         print(y)
         print(z)
```

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

```
In [75]: x.intersection(y) #Only common element present in both side
```

```
Out[75]: {5, 6}
```

```
In [76]: y.intersection(z)
```

```
Out[76]: {8, 9}
```

```
In [78]: y & x
```

```
Out[78]: {5, 6}
```

```
In [79]: z&y
```

```
Out[79]: {8, 9}
```

DIFFERENCE(-)

```
In [81]: a={1,2,3,4,5}
         b={6,7,8,9,10}
         c={20,30,40,50,60}
```

```
In [82]: a.difference(b) # returns items that are in the first set but not in second set
```

```
Out[82]: {1, 2, 3, 4, 5}
```

```
In [83]: b.difference(c)

Out[83]: {6, 7, 8, 9, 10}

In [84]: c-a

Out[84]: {20, 30, 40, 50, 60}

In [85]: a-b

Out[85]: {1, 2, 3, 4, 5}
```

SYMMETRIC DIFFERENCE(^)

```
In [86]: A={1,2,3,4,5}
         B={4,5,6,7,8}

In [87]: A.symmetric_difference(B) #returns items that are in either set but not in both (re

Out[87]: {1, 2, 3, 6, 7, 8}

In [88]: B.symmetric_difference(a)

Out[88]: {1, 2, 3, 6, 7, 8}
```

Subset,superset,disjoint

```
In [97]: A = {1,2,3,4,5,6,7,8,9}
         B = {3,4,5,6,7,8}
         C = {10,20,30,40}

In [98]: B.issuperset(C) # if it contain all element of another set

Out[98]: False

In [99]: A.issuperset(B)

Out[99]: True

In [100…  A.issubset(B) # if all elements are present in another set

Out[100…  False

In [101…  B.issubset(A)

Out[101…  True

In [103…  B.isdisjoint(A) # if they have not common element
```

Out[103…

```
False
```

In [104…

```
a.isdisjoint(c)
```

Out[104…

```
True
```

In [105…

```
c.isdisjoint(B)
```

Out[105…

```
True
```

Set membership

In [106…

```
myset={'ONE','TWO','THREE','FOUR','FIVE'}
myset
```

Out[106…

```
{'FIVE', 'FOUR', 'ONE', 'THREE', 'TWO'}
```

In [107…

```
'ONE' in myset
```

Out[107…

```
True
```

In [108…

```
'TEN' in myset
```

Out[108…

```
False
```

In [110…

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

```
FIVE is present in the set
```

Loop through a set

In [111…

```
myset
```

Out[111…

```
{'FIVE', 'FOUR', 'ONE', 'THREE', 'TWO'}
```

In [112…

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

```
THREE
FIVE
ONE
FOUR
TWO
```

In [113…

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

```
(0, 'THREE')
(1, 'FIVE')
(2, 'ONE')
(3, 'FOUR')
(4, 'TWO')
```

In [ ]:

```

```

In [ ]:

```

```