Sets

*Set Creation

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

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

In [2]:
```
len(s)
```

Out[2]:
```
5
```

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

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

In [4]:
```
s1={1.79,2.08,3.99,4.56,5.45} # Set of float numbers
s1
```

Out[4]:
```
{1.79, 2.08, 3.99, 4.56, 5.45}
```

In [5]:
```
s2={'sai','madhu','krishna'} # Set of Strings
s2
```

Out[5]:
```
{'krishna', 'madhu', 'sai'}
```

In [6]:
```
s3={10,20,'hola',(11,22,33)} # Mixed datatypes
s3
```

Out[6]:
```
{(11, 22, 33), 10, 20, 'hola'}
```

In [8]:
```
s3={10,20,'hola',[11,22,33]}
s3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 s3={10,20,'hola',[11,22,33]}
      2 s3

TypeError: unhashable type: 'list'
```

In [9]:
```
s4 = set() # Create an empty set
print(type(s4))
```

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

In [10]:
```
s1 = set(('one' , 'two' , 'three' , 'four'))
s1
```

Out[10]:
```
{'four', 'one', 'three', 'two'}
```

Loop through a Set

In [13]:

```
s= {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
for i in s:
    print(i)
```

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

In [14]:

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

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

Set Membership

In [15]:

```
s
```

Out[15]:

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [16]:

```
'one' in s # check if "one" exist in the set
```

Out[16]:

```
True
```

In [17]:

```
'ten' in s # check if 'ten' exist in the set
```

Out[17]:

```
False
```

In [19]:

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

```
three is present in the set
```

In [22]:

```
if 'eleven' in s:
    print('eleven is present in the set')
else:
    print('eleven is not present in the set')
```

```
eleven is not present in the set
```

Add & Remove Items

In [ ]:

```

```

In [23]:

```
s
```

Out[23]:

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [24]:

```
s.add('nine') # Add item to a set using add() method
s
```

Out[24]:

```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [26]:

```
s.update(['TEN','ELEVEN','TWELVE'])
s
```

Out[26]:

```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'nine',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [27]:

```
s.remove('nine')
s
```

Out[27]:

```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [28]:

```
s.discard('TEN') # remove item from a set using discard() method
s
```

```
Out[28]: {'ELEVEN',
          'TWELVE',
          'eight',
          'five',
          'four',
          'one',
          'seven',
          'six',
          'three',
          'two'}
```

```
In [29]: s.clear()
         s
```

```
Out[29]: set()
```

```
In [30]: del s
         s
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[30], line 2
      1 del s
----> 2 s

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

Copy Set

```
In [31]: s = {'one','two','three','four','five','six','seven','eight'}
         s
```

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

```
In [32]: s1 = s
         s1
```

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

```
In [33]: id(s), id(s1)
```

```
Out[33]: (2292586929216, 2292586929216)
```

```
In [39]: s = s1.copy() # Create a copy of the list
         s
```

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

```
In [40]: id(s)
```

```
Out[40]: 2292586933696
```

```
In [41]: s.add('nine')
         s
```

```
Out[41]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}

In [42]: s1  # s1 will be also impacted as it is pointing to the same Set

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

In [43]: s  # Copy of the set won't be impacted due to changes made on the original S

Out[43]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

Set Operation

- Union

```
In [45]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}

In [46]: a|b

Out[46]: {1, 2, 3, 4, 5, 6, 7, 8}

In [47]: a.union(b)

Out[47]: {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]: """
         Updates the set calling the update() method with union of a , b & c.
         For below example Set A will be updated with union of a,b & c.
         """
         a.update(b,c)
         a

Out[52]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Intersection

```
In [53]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}

In [54]: a&b

Out[54]: {4, 5}

In [55]: a.intersection(b) intersection of a and b
```

```
  Cell In[55], line 1
    a.intersection(b) intersection of a and b
                      ^
SyntaxError: invalid syntax
```

```
In [56]: a.intersection_update(b)
         a
```

```
Out[56]: {4, 5}
```

Difference

```
In [57]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
```

```
In [58]: a-b
```

```
Out[58]: {1, 2, 3}
```

```
In [59]: a.difference(b)
```

```
Out[59]: {1, 2, 3}
```

```
In [60]: b-a
```

```
Out[60]: {6, 7, 8}
```

```
In [61]: b.difference(a)
```

```
Out[61]: {6, 7, 8}
```

Symmetric Difference

```
In [62]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
```

```
In [63]: a ^ b
```

```
Out[63]: {1, 2, 3, 6, 7, 8}
```

```
In [64]: a.symmetric_difference(b)
```

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

Subset , Superset & Disjoint

```
In [65]: a = {1,2,3,4,5,6,7,8,9}
         b = {3,4,5,6,7,8}
         c = {10,20,30,40}
```

In [66]:
```
b.issubset(a)
```
Out[66]:
```
True
```

In [67]:
```
a.issuperset(b)
```
Out[67]:
```
True
```

In [68]:
```
c.isdisjoint(a)
```
Out[68]:
```
True
```

In [69]:
```
b.isdisjoint(a)
```
Out[69]:
```
False
```

In [70]:
```
# Other Builtin functions
```

In [71]:
```
a
```
Out[71]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [72]:
```

```
Out[72]:
```
45
```

In [73]:
```
max(a)
```
Out[73]:
```
9
```

In [74]:
```
min(a)
```
Out[74]:
```
1
```

In [75]:
```
len(a)
```
Out[75]:
```
9
```

In [76]:
```
list(enumerate(a))
```
Out[76]:
```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```

In [77]:
```
d=sorted(a,reverse=True)
d
```
Out[77]:
```
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [78]:
```
sorted(d)
```
Out[78]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Dictionary

In [ ]:
```
#  Create dictionary
```

In [79]:
```
d= dict()
d
```

Out[79]:
```
{}
```

In [80]:
```
d={}
d
```

Out[80]:
```
{}
```

In [81]:
```
d={1:'one' , 2:'two' , 3:'three'}
d
```

Out[81]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [82]:
```
d=dict({1:'one',2:'two',3:'three'})
d
```

Out[82]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [83]:
```
d= {'A':'one' , 'B':'two' , 'C':'three'}
d
```

Out[83]:
```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

In [84]:
```
d= {1:'one' , 'A':'two' , 3:'three'}
d
```

Out[84]:
```
{1: 'one', 'A': 'two', 3: 'three'}
```

In [85]:
```
d.keys()
```

Out[85]:
```
dict_keys([1, 'A', 3])
```

In [86]:
```
d.values()
```

Out[86]:
```
dict_values(['one', 'two', 'three'])
```

In [87]:
```
d.items()
```

Out[87]:
```
dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])
```

In [89]:
```
d={1:'one',2:'two','A':['sai' ,'kumar','hari']}
d
```

Out[89]:
```
{1: 'one', 2: 'two', 'A': ['sai', 'kumar', 'hari']}
```

In [90]:
```
d={1:'one',2:'two','A':['sai' ,'kumar','hari'],'B':('Bat','cat','hat')}
d
```

Out[90]:
```
{1: 'one', 2: 'two', 'A': ['sai', 'kumar', 'hari'], 'B': ('Bat', 'cat', 'hat')}
```

In [91]:
```
d={1:'one',2:'two','A':{'Name':'saikumar','Age':20},'B':('bat','cat','hat')}
```

```
d
```

Out[91]:
```
{1: 'one',
 2: 'two',
 'A': {'Name': 'saikumar', 'Age': 20},
 'B': ('bat', 'cat', 'hat')}
```

In [92]:
```
keys = {'a','b','c','d'}
d3=dict.fromkeys(keys)
d3
```

Out[92]:
```
{'a': None, 'd': None, 'b': None, 'c': None}
```

In [93]:
```
keys = {'a','b','c','d'}
value= 10
d3=dict.fromkeys(keys,value)
d3
```

Out[93]:
```
{'a': 10, 'd': 10, 'b': 10, 'c': 10}
```

In [94]:
```
keys = {'a','b','c','d'}
value= [10,20,30]
d3=dict.fromkeys(keys,value)
d3
```

Out[94]:
```
{'a': [10, 20, 30], 'd': [10, 20, 30], 'b': [10, 20, 30], 'c': [10, 20, 30]}
```

In [95]:
```
value.append(40)
d3
```

Out[95]:
```
{'a': [10, 20, 30, 40],
 'd': [10, 20, 30, 40],
 'b': [10, 20, 30, 40],
 'c': [10, 20, 30, 40]}
```

Accessing Items

In [96]:
```
d={1:'one',2:'two',3:'three',4:'four'}
d
```

Out[96]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [97]:
```
d[1]
```

Out[97]:
```
'one'
```

In [99]:
```
d.get(1)
```

Out[99]:
```
'one'
```

In [101…
```
d1={'Name':'Madhu','ID':85502,'DOB':1985,'job':'analyst'}
d1
```

Out[101…
```
{'Name': 'Madhu', 'ID': 85502, 'DOB': 1985, 'job': 'analyst'}
```

In [102…
```
d1['Name']
```

```
Out[102…   'Madhu'

In [103…   d1.get('job')

Out[103…   'analyst'

In [104…   # Add , Remove & Change Items

In [114…   d1 = {'Name':'madhu' , 'ID': 85502 , 'DOB': 1985 , 'Address' : 'Hilsinki'}
           d1

Out[114…   {'Name': 'madhu', 'ID': 85502, 'DOB': 1985, 'Address': 'Hilsinki'}

In [116…   d1['DOB'] = 1992 # Changing Dictionary Items
           d1['Address'] = 'Delhi'
           d1

Out[116…   {'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Delhi'}

In [111…   d1 = {'DOB':1995}
           d1.update(d1)
           d1

Out[111…   {'DOB': 1995}

In [118…   d1['Job'] = 'Analyst' # Adding items in the dictionary
           d1

Out[118…   {'Name': 'madhu',
            'ID': 85502,
            'DOB': 1992,
            'Address': 'Delhi',
            'Job': 'Analyst'}

In [117…   d1

Out[117…   {'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Delhi'}

In [119…   d1.pop('Job') # Removing items in the dictionary using Pop method
           d1

Out[119…   {'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Delhi'}

In [120…   d1.popitem()

Out[120…   ('Address', 'Delhi')

In [121…   d1

Out[121…   {'Name': 'madhu', 'ID': 85502, 'DOB': 1992}

In [122…   del[d1['ID']]
           d1

Out[122…   {'Name': 'madhu', 'DOB': 1992}
```

In [123…
```
d1.clear()
d1
```

Out[123…
```
{}
```

In [124…
```
del d1
d1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[124], line 2
      1 del d1
----> 2 d1

NameError: name 'd1' is not defined
```

Copy Dictionary

In [125…
```
d = {'Name':'madhu' , 'ID': 85502 , 'DOB': 1992 , 'Address' : 'Hilsinki'}
d
```

Out[125…
```
{'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Hilsinki'}
```

In [126…
```
d1 = d
```

In [127…
```
id(d),id(d1)
```

Out[127…
```
(2292600038720, 2292600038720)
```

In [128…
```
d2 = d.copy()
```

In [129…
```
id(d2)
```

Out[129…
```
2292599698752
```

In [130…
```
d['Address'] = 'hyderabad'
```

In [131…
```
d
```

Out[131…
```
{'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'hyderabad'}
```

In [132…
```
d1
```

Out[132…
```
{'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'hyderabad'}
```

In [133…
```
d2
```

Out[133…
```
{'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Hilsinki'}
```

Loop Through a Dictionary

In [135…

```
d1={'Name': 'madhu', 'ID': 85502, 'DOB': 1992, 'Address': 'Hilsinki','Job': 'ana
d1
```

Out[135…

```
{'Name': 'madhu',
 'ID': 85502,
 'DOB': 1992,
 'Address': 'Hilsinki',
 'Job': 'analyst'}
```

In [136…

```
for i in d1:
    print(i,':',d1[i])
```

```
Name : madhu
ID : 85502
DOB : 1992
Address : Hilsinki
Job : analyst
```

In [137…

```
for i in d1:
    print(d1[i])
```

```
madhu
85502
1992
Hilsinki
analyst
```

Dictionary membership

In [139…

```
d1= {'Name':'Saikumar','ID':74101, 'DOB':1999, 'Job':'Analyst'}
d1
```

Out[139…

```
{'Name': 'Saikumar', 'ID': 74101, 'DOB': 1999, 'Job': 'Analyst'}
```

In [140…

```
'Name' in d1
```

Out[140…

```
True
```

In [ ]:

```
'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: