Sets

Set creaton

In [6]:

```
mylist = {1,2,3,4,5}
mylist
```

Out[6]:

```
{1, 2, 3, 4, 5}
```

In [8]:

```
len(mylist)
```

Out[8]:

```
5
```

In [10]:

```
my_set = {1,1,2,3,4,5,5}
my_set
```

Out[10]:

```
{1, 2, 3, 4, 5}
```

In [12]:

```
myset1 = {1.2,3.3,3.7,5.7,8.8}
myset1
```

Out[12]:

```
{1.2, 3.3, 3.7, 5.7, 8.8}
```

In [17]:

```
myset2 = {'kavya' ,'sreya' ,'arshiya'}
myset2
```

Out[17]:

```
{'arshiya', 'kavya', 'sreya'}
```

In [21]:

```
myset3 = {1,2,"Hi", (1,2,3)}
myset3
```

Out[21]:

```
{(1, 2, 3), 1, 2, 'Hi'}
```

In [25]:

```
myset4 = set()
print(type(myset4))
```

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

In [27]:

```
my_set1 = set(('shiva','kavya','akshitha'))
my_set1
```

Out[27]:

```
{'akshitha', 'kavya', 'shiva'}
```

Loop through a set

In [30]:

```
myset = {'one','two','three','four'}
for i in myset:
    print(i)
```

```
three
one
four
two
```

In [32]:

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

```
(0, 'three')
(1, 'one')
(2, 'four')
(3, 'two')
```

set membership

In [35]:

```
myset
```

Out[35]:

```
{'four', 'one', 'three', 'two'}
```

In [39]:

```
'one' in myset
```

Out[39]:

```
True
```

In [41]:

```
'four' in myset
```

Out[41]:

```
True
```

In [43]:

```
'five' in myset
```

Out[43]:

```
False
```

In [47]:

```
'three' in myset
```

Out[47]:

```
True
```

In [49]:

```python
'eight' in myset
```

Out[49]:

```
False
```

Add & remove items

In [24]:

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

In [26]:

```python
myset
```

Out[26]:

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

In [28]:

```python
myset.add('NINE') # Add item to a set using add() method
myset
```

Out[28]:

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

In [30]:

```python
myset.update(['TEN' , 'ELEVEN' , 'TWELVE']) # Add multiple item to a set using
myset
```

Out[30]:

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

In [32]:

```python
myset.remove('NINE') # remove item in a set using remove() method
myset
```

Out[32]:

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

In [34]:

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

Out[34]:

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

In [36]:

```
myset.clear()
myset
```

Out[36]:

```
set()
```

In [38]:

```
del myset
myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[38], line 2
      1 del myset # Delete the set object
----> 2 myset

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

copy set

In [43]:

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

Out[43]:

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

In [45]:

```
myset1 = myset
myset
```

Out[45]:

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

In [47]:

```
id(myset) , id(myset1)
```

Out[47]:

```
(1436738559488, 1436738559488)
```

In [49]:

```
my_set = myset.copy()
my_set
```

Out[49]:

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

In [51]:

```
id(my_set)
```

Out[51]:

```
1436738561504
```

In [53]:

```
myset.add('nine')
myset
```

Out[53]:

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

In [55]:

```
myset1
```

Out[55]:

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

In [57]:

```
my_set
```

Out[57]:

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

SET OPERATION

Union

In [61]:

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [63]:

```
A | B
```

Out[63]:

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

In [65]:

```
A.union(B)
```

Out[65]:

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

In [67]:

```
A.union(B, C)
```

Out[67]:

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

intersection

In [70]:

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [72]:

```
A & B
```

Out[72]:

```
{4, 5}
```

In [74]:

```
A.intersection(B) Intersection of A and B
```

```
  Cell In[74], line 1
    A.intersection(B) Intersection of A and B
                      ^
SyntaxError: invalid syntax
```

difference

In [77]:

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [79]:

```
A - B
```

Out[79]:

```
{1, 2, 3}
```

In [81]:

```
A.difference(B)
```

Out[81]:

```
{1, 2, 3}
```

In [83]:

```
A.difference(B)
```

Out[83]:

```
{1, 2, 3}
```

In [85]:

```
B- A
```

Out[85]:

```
{6, 7, 8}
```

In [87]:

```
B.difference(A)
```

Out[87]:

```
{6, 7, 8}
```

symetric difference

In [90]:

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [92]:

```
A ^ B
```

Out[92]:

```
{1, 2, 3, 6, 7, 8}
```

In [94]:

```
A.symmetric_difference(B)
```

Out[94]:

```
{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 [99]:

```
B.issubset(A)
```

Out[99]:

```
True
```

In [101]:

```
A.issuperset(B)
```

Out[101]:

```
True
```

In [103]:

```
C.isdisjoint(A)
```

Out[103]:

```
True
```

In [105]:

```
B.isdisjoint(A)
```

Out[105]:

```
False
```

In [107]:

```
B.isdisjoint(A)
```

Out[107]:

```
False
```

Other Builtin functions

In [110]:

```
A
```

Out[110]:

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

In [112]:

```
sum(A)
```

Out[112]:

```
45
```

In [114]:

```
max(A)
```

Out[114]:

```
9
```

In [116]:

```
min(A)
```

Out[116]:

```
1
```

In [118]:

```
len(A)
```

Out[118]:

```
9
```

In [120]:

```
list(enumerate(A))
```

Out[120]:

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

In [122]:

```
D= sorted(A,reverse=True)
D
```

Out[122]:

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

In [124]:

```
sorted(D)
```

Out[124]:

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

DICTONARY

Create Dictionary

In [130]:

```
mydict = dict()
mydict
```

Out[130]:

```
{}
```

In [134]:

```
mydict = {}
mydict
```

Out[134]:

```
{}
```

In [136]:

```
mydict = {1:'one' , 2:'two' , 3:'three'}
mydict
```

Out[136]:

```
{1: 'one', 2: 'two', 3: 'three'}
```

In [138]:

```
mydict = dict({1:'one' , 2:'two' , 3:'three'})
mydict
```

Out[138]:

```
{1: 'one', 2: 'two', 3: 'three'}
```

In [140]:

```
mydict = {'A':'one' , 'B':'two' , 'C':'three'}
mydict
```

Out[140]:

```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

In [142]:

```
mydict = {1:'one' , 'A':'two' , 3:'three'}
mydict
```

Out[142]:

```
{1: 'one', 'A': 'two', 3: 'three'}
```

In [144]:

```
mydict.keys()
```

Out[144]:

```
dict_keys([1, 'A', 3])
```

In [146]:

```
mydict.values()
```

Out[146]:

```
dict_values(['one', 'two', 'three'])
```

In [148]:

```
mydict.items()
```

Out[148]:

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

In [150]:

```
mydict = {1:'one' , 2:'two' , 'A':['asif' , 'john' , 'Maria']}
mydict
```

Out[150]:

```
{1: 'one', 2: 'two', 'A': ['asif', 'john', 'Maria']}
```

In [157]:

```
 mydict = {1:'one' , 2:'two' , 'A':['asif' , 'john' , 'Maria'], 'B':('Bat' , 'cat' , 'hat')}
mydict
```

Out[157]:

```
{1: 'one',
 2: 'two',
 'A': ['asif', 'john', 'Maria'],
 'B': ('Bat', 'cat', 'hat')}
```

In [159]:

```
keys = {'a' , 'b' , 'c' , 'd'}
mydict3 = dict.fromkeys(keys)
mydict3
```

Out[159]:

```
{'a': None, 'd': None, 'b': None, 'c': None}
```

In [161]:

```
keys = {'a' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict3 = dict.fromkeys(keys , value)
mydict3
```

Out[161]:

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

In [163]:

```
value.append(40)
mydict3
```

Out[163]:

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

Accessing Items

In [166]:

```
mydict = {1:'one' , 2:'two' , 3:'three' , 4:'four'}
mydict
```

Out[166]:

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [168]:

```
mydict[1]
```

Out[168]:

```
'one'
```

In [170]:

```
mydict.get(1)
```

Out[170]:

```
'one'
```

In [172]:

```
mydict1 = {'Name':'kavya' , 'ID': 74123 , 'DOB': 2004 , 'job' :'Analyst'}
mydict1
```

Out[172]:

```
{'Name': 'kavya', 'ID': 74123, 'DOB': 2004, 'job': 'Analyst'}
```

In [174]:

```
mydict1['Name']
```

Out[174]:

```
'kavya'
```

In [176]:

```
mydict1['job']
```

Out[176]:

```
'Analyst'
```

Add, Remove & Change Items

In [181]:

```
mydict1 = {'Name':'kavya' , 'ID': 74123 , 'DOB': 2004 , 'job' :'Analyst'}
mydict1
```

Out[181]:

```
{'Name': 'kavya', 'ID': 74123, 'DOB': 2004, 'job': 'Analyst'}
```

In [183]:

```
dict1 = {'DOB':2004}
mydict1.update(dict1)
mydict
```

Out[183]:

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [185]:

```
mydict1['Job'] = 'Analyst'
mydict
```

Out[185]:

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [187]:

```
mydict1['Job'] = 'Analyst'
mydict1
```

Out[187]:

```
{'Name': 'kavya', 'ID': 74123, 'DOB': 2004, 'job': 'Analyst', 'Job': 'Analyst'}
```

In [189]:

```
mydict1.popitem()
```

Out[189]:

```
('Job', 'Analyst')
```

In [191]:

```
mydict1
```

Out[191]:

```
{'Name': 'kavya', 'ID': 74123, 'DOB': 2004, 'job': 'Analyst'}
```

Copy Dictionary

In [200]:

```
mydict = {'Name':'kavya' , 'ID': 12345 , 'DOB': 2004 , 'Address' : 'siddipet'}
mydict
```

Out[200]:

```
{'Name': 'kavya', 'ID': 12345, 'DOB': 2004, 'Address': 'siddipet'}
```

In [202]:

```
mydict1 = mydict
```

In [206]:

```
id(mydict) , id(mydict1)
```

Out[206]:

```
(1436752062912, 1436752062912)
```

In [208]:

```
mydict2 = mydict.copy()
```

In [210]:

```
id(mydict2)
```

Out[210]:

```
1436752058112
```

In [212]:

```
mydict['Address'] = 'siddipet'
```

In [214]:

```
mydict
```

Out[214]:

```
{'Name': 'kavya', 'ID': 12345, 'DOB': 2004, 'Address': 'siddipet'}
```

Loop through a Dictionary

In [219]:

```
mydict1 = {'Name':'kavya' , 'ID': 12345 , 'DOB': 2004 , 'Address' : 'siddipet'}
```

In [221]:

```
mydict
```

Out[221]:

```
{'Name': 'kavya', 'ID': 12345, 'DOB': 2004, 'Address': 'siddipet'}
```

In [229]:

```
for i in mydict1:
    print(mydict1[i])
```

```
kavya
12345
2004
siddipet
```

RANGE

In [232]:

```
range(5)
```

Out[232]:

```
range(0, 5)
```

In [234]:

```
list(range(0,5))
```

Out[234]:

```
[0, 1, 2, 3, 4]
```

In [236]:

```
range(10,20)
```

Out[236]:

```
range(10, 20)
```

In [238]:

```
list(range(10,20))
```

Out[238]:

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [240]:

```
range(10,20,3)
```

Out[240]:

```
range(10, 20, 3)
```

In [242]:

```
list(range(10,20,5))
```

Out[242]:

```
[10, 15]
```

In [244]:

```
range(0,10,3,2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[244], line 1
----> 1 range(0,10,3,2)

TypeError: range expected at most 3 arguments, got 4
```

In [246]:

```
r = range(0,11,5)
r
```

Out[246]:

```
range(0, 11, 5)
```

In [248]:

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

```
0
5
10
```

In [250]:

```
list(r)
```

Out[250]:

```
[0, 5, 10]
```

advanced slicing

In [253]:

```
l = ['a', 'b', 'c', 1, 2.3, 45, True, 1+2j]
```

In [255]:

```
l
```

Out[255]:

```
['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]
```

In [257]:

```
l[:]
```

Out[257]:

```
['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]
```

In [259]:

```
l[3:]
```

Out[259]:

```
[1, 2.3, 45, True, (1+2j)]
```

In [261]:

```
l
```

Out[261]:

```
['a', 'b', 'c', 1, 2.3, 45, True, (1+2j)]
```

In [263]:

```
l[5]
```

Out[263]:

```
45
```

In [ ]:

```

```