SET

1) Unordered & Unindexed collection of items.

2) Set elements are unique. Duplicate elements are not allowed.

3) Set elements are immutable (cannot be changed).

4) Set itself is mutable. We can add or remove items from it.

SET CREATION

```
In [2]: myset = {1,2,3,4,5} # Set of numbers
        myset

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

```
In [3]: len(myset) #Length of the set

Out[3]: 5
```

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

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

```
In [5]: myset1 = {1.79,2.08,3.99,4.56,5.45} # Set of float numbers
        myset1

Out[5]: {1.79, 2.08, 3.99, 4.56, 5.45}
```

```
In [8]: myset2 = {'Sidra' , 'Maimona' , 'Zoha'} # Set of Strings
        myset2

Out[8]: {'Maimona', 'Sidra', 'Zoha'}
```

```
In [9]: myset3 = {10,20, "Sidra", (11, 22, 32)} # Mixed datatypes
        myset3

Out[9]: {(11, 22, 32), 10, 20, 'Sidra'}
```

```
In [11]: myset3 = {10,20, "Sidra", [11, 22, 32]} # set doesn't allow mutable items like li
         myset3

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         Input In [11], in <cell line: 1>()
         ----> 1 myset3 = {10,20, "Sidra", [11, 22, 32]} # set doesn't allow mutable items like li
               2 myset3

         TypeError: unhashable type: 'list'
```

```
In [12]: myset4 = set() # Create an empty set
         print(type(myset4))

         <class 'set'>
```

In [13]:

```
my_set1 = set(('one' , 'two' , 'three' , 'four'))
my_set1
```

Out[13]:

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

LOOP THROUGH A SET

In [15]:

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

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

In [16]:

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

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

SET MEMBERSHIP

In [17]:

```
myset
```

Out[17]:

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

In [18]:

```
'one' in myset # Check if 'one' exist in the set
```

Out[18]:

```
True
```

In [19]:

```
'ten' in myset
```

Out[19]:

```
False
```

In [21]:

```
if 'three' in myset:
    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 myset:
    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 [23]: myset
```

```
Out[23]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [24]: myset.add('NINE') # Add item to a set using add() method
         myset
```

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

```
In [25]: myset.update(['TEN' , 'ELEVEN' , 'TWELVE']) # Add multiple item to a set using
         myset
```

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

```
In [26]: myset.remove('NINE') # remove item in a set using remove() method
         myset
```

```
Out[26]: {'ELEVEN',
          'TEN',
          'TWELVE',
          'eight',
          'five',
          'four',
          'one',
          'seven',
          'six',
          'three',
          'two'}
```

```
In [27]: myset.discard('TEN') # remove item from a set using discard() method
         myset
```

```
Out[27]: {'ELEVEN',
          'TWELVE',
          'eight',
          'five',
          'four',
          'one',
          'seven',
          'six',
          'three',
          'two'}
```

```
In [28]: myset.clear() # Delete all items in a set
         myset
```

```
Out[28]: set()
```

```
In [29]: del myset # Delete the set object
         myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [29], in <cell line: 2>()
      1 del myset # Delete the set object
----> 2 myset

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

COPY SET

```
In [30]: myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
         myset

Out[30]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [31]: myset1 = myset # Create a new reference "myset1"
         myset1

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

```
In [32]: id(myset) , id(myset1)

Out[32]: (2601859198304, 2601859198304)
```

```
In [33]: my_set = myset.copy() # Create a copy of the list
         my_set

Out[33]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [34]: id(my_set)

Out[34]: 2601860493824
```

```
In [35]: myset.add('nine')
         myset

Out[35]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [36]: myset1

Out[36]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [37]: my_set

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

SET OPERATIONS

UNION

```
In [38]: A = {1,2,3,4,5}
         B = {4,5,6,7,8}
         C = {8,9,10}
```

```
In [39]: A | B

Out[39]: {1, 2, 3, 4, 5, 6, 7, 8}

In [40]: A.union(B)

Out[40]: {1, 2, 3, 4, 5, 6, 7, 8}

In [41]: A.union(B, C)

Out[41]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [42]: A.update(B,C)
         A

Out[42]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

INTERSECTION

```
In [43]: A = {1,2,3,4,5}
         B = {4,5,6,7,8}

In [44]: A & B

Out[44]: {4, 5}

In [45]: A.intersection(B) Intersection of A and B

           Input In [45]
             A.intersection(B) Intersection of A and B
                                            ^
         SyntaxError: invalid syntax


In [46]: A.intersection_update(B)
         A

Out[46]: {4, 5}
```

DIFFERENCE

```
In [47]: A = {1,2,3,4,5}
         B = {4,5,6,7,8}

In [48]: A - B

Out[48]: {1, 2, 3}

In [49]: A.difference(B)

Out[49]: {1, 2, 3}

In [50]: B - A

Out[50]: {6, 7, 8}

In [51]: B.difference(A)

Out[51]: {6, 7, 8}
```

```
In [52]: B.difference(A)

Out[52]: {6, 7, 8}
```

SYMMETRIC DIFFERENCE

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

```
In [54]: A ^ B

Out[54]: {1, 2, 3, 6, 7, 8}
```

```
In [55]: A.symmetric_difference(B)

Out[55]: {1, 2, 3, 6, 7, 8}
```

```
In [56]: A.symmetric_difference_update(B)
         A

Out[56]: {1, 2, 3, 6, 7, 8}
```

SUBSET, SUPERSET & DISJOINT

```
In [57]: A = {1,2,3,4,5,6,7,8,9}
         B = {3,4,5,6,7,8}
         C = {10,20,30,40}
```

```
In [58]: B.issubset(A)

Out[58]: True
```

```
In [60]: A.issuperset(B)

Out[60]: True
```

```
In [61]: C.isdisjoint(A)

Out[61]: True
```

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

Out[62]: False
```

OTHER BUILT-IN FUNCTIONS

```
In [63]: A

Out[63]: {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In [64]: sum(A)

Out[64]: 45
```

```
In [65]: max(A)

Out[65]: 9
```

```
In [66]: min(A)

Out[66]: 1

In [67]: len(A)

Out[67]: 9

In [68]: list(enumerate(A))

Out[68]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]

In [69]: D= sorted(A,reverse=True)
         D

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

In [70]: sorted(D)

Out[70]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

DICTIONARY

. Dictionary is a mutable data type in Python.

. A python dictionary is a collection of key and value pairs separated by a colon (:) & enclosed in curly braces {}.

. Keys must be unique in a dictionary, duplicate values are allowed.

CREATE DICTIONARY

```
In [71]: mydict = dict() # empty dictionary
         mydict

Out[71]: {}

In [72]: mydict = {} # empty dictionary
         mydict

Out[72]: {}

In [73]: mydict = {1:'one' , 2:'two' , 3:'three'} # dictionary with integer keys
         mydict

Out[73]: {1: 'one', 2: 'two', 3: 'three'}

In [74]: mydict = dict({1:'one' , 2:'two' , 3:'three'}) # Create dictionary using dict()
         mydict

Out[74]: {1: 'one', 2: 'two', 3: 'three'}

In [75]: mydict = {'A':'one' , 'B':'two' , 'C':'three'} # dictionary with character keys
         mydict

Out[75]: {'A': 'one', 'B': 'two', 'C': 'three'}

In [76]: mydict = {1:'one' , 'A':'two' , 3:'three'} # dictionary with mixed keys
         mydict

Out[76]: {1: 'one', 'A': 'two', 3: 'three'}
```

```
In [77]: mydict.keys()

Out[77]: dict_keys([1, 'A', 3])
```

```
In [78]: mydict.values()

Out[78]: dict_values(['one', 'two', 'three'])
```

```
In [79]: mydict.items()

Out[79]: dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])
```

```
In [80]: mydict = {1:'one' , 2:'two' , 'A':['asif' , 'john' , 'Maria']} # dictionary with
         mydict

Out[80]: {1: 'one', 2: 'two', 'A': ['asif', 'john', 'Maria']}
```

```
In [87]: mydict = {1:'one' , 2:'two' , 'A':['asif' , 'john' , 'Maria'], 'B':('Bat' , 'cat' , 'mat')}
         mydict

Out[87]: {1: 'one',
          2: 'two',
          'A': ['asif', 'john', 'Maria'],
          'B': ('Bat', 'cat', 'mat')}
```

```
In [90]: mydict = {1:'one' , 2:'two' , 'A':{'Name':'asif' , 'Age' :20}, 'B':('Bat' , 'cat', 'mat')}
         mydict

Out[90]: {1: 'one',
          2: 'two',
          'A': {'Name': 'asif', 'Age': 20},
          'B': ('Bat', 'cat', 'mat')}
```

```
In [91]: keys = {'a' , 'b' , 'c' , 'd'}
         mydict3 = dict.fromkeys(keys) # Create a dictionary from a sequence of keys
         mydict3

Out[91]: {'c': None, 'b': None, 'd': None, 'a': None}
```

```
In [92]: keys = {'a' , 'b' , 'c' , 'd'}
         value = 10
         mydict3 = dict.fromkeys(keys , value) # Create a dictionary from a sequence of
         mydict3

Out[92]: {'c': 10, 'b': 10, 'd': 10, 'a': 10}
```

```
In [93]: keys = {'a' , 'b' , 'c' , 'd'}
         value = [10,20,30]
         mydict3 = dict.fromkeys(keys , value) # Create a dictionary from a sequence of
         mydict3

Out[93]: {'c': [10, 20, 30], 'b': [10, 20, 30], 'd': [10, 20, 30], 'a': [10, 20, 30]}
```

```
In [94]: value.append(40)
         mydict3

Out[94]: {'c': [10, 20, 30, 40],
          'b': [10, 20, 30, 40],
          'd': [10, 20, 30, 40],
          'a': [10, 20, 30, 40]}
```

ACCESSING ITEMS

```
In [95]: mydict = {1:'one' , 2:'two' , 3:'three' , 4:'four'}
         mydict
```

```
Out[95]: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
In [96]: mydict[1]
```

```
Out[96]: 'one'
```

```
In [97]: mydict.get(1)
```

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

```
In [103]: mydict1 = {'Name':'Sidra' , 'ID': 74123 , 'DOB': 1991 , 'job' :'Analyst'}
          mydict1
```

```
Out[103]: {'Name': 'Sidra', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}
```

```
In [104]: mydict1['Name']
```

```
Out[104]: 'Sidra'
```

```
In [105]: mydict1.get('job')
```

```
Out[105]: 'Analyst'
```

ADD , REMOVE & CHANGE ITEMS

```
In [107]: mydict1 = {'Name':'Sidra' , 'ID': 12345 , 'DOB': 2004 , 'Address' : 'Chicago'}
          mydict1
```

```
Out[107]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Address': 'Chicago'}
```

```
In [108]: mydict1['DOB'] = 2001 # Changing Dictionary Items
          mydict1['Address'] = 'Hyderabad'
          mydict1
```

```
Out[108]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2001, 'Address': 'Hyderabad'}
```

```
In [109]: dict1 = {'DOB':2000}
          mydict1.update(dict1)
          mydict1
```

```
Out[109]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2000, 'Address': 'Hyderabad'}
```

```
In [110]: mydict1['Job'] = 'Analyst' # Adding items in the dictionary
          mydict1
```

```
Out[110]: {'Name': 'Sidra',
           'ID': 12345,
           'DOB': 2000,
           'Address': 'Hyderabad',
           'Job': 'Analyst'}
```

```
In [111]: mydict1.pop('Job') # Removing items in the dictionary using Pop method
          mydict1
```

```
Out[111]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2000, 'Address': 'Hyderabad'}
```

```
In [112]: mydict1.popitem() # A random item is removed
```

```
Out[112]: ('Address', 'Hyderabad')
```

```
In [113]: mydict1

Out[113]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2000}

In [114]: del[mydict1['ID']] # Removing item using del method
          mydict1

Out[114]: {'Name': 'Sidra', 'DOB': 2000}

In [115]: mydict1.clear() # Delete all items of the dictionary using clear method
          mydict1

Out[115]: {}

In [116]: del mydict1 # Delete the dictionary object
          mydict1

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          Input In [116], in <cell line: 2>()
                1 del mydict1 # Delete the dictionary object
          ----> 2 mydict1

          NameError: name 'mydict1' is not defined
```

COPY DICTIONARY

```
In [117]: mydict = {'Name':'Sidra' , 'ID': 12345 , 'DOB': 2004 , 'Address' : 'Chicago'}
          mydict

Out[117]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Address': 'Chicago'}

In [118]: mydict1 = mydict # Create a new reference "mydict1"

In [119]: id(mydict) , id(mydict1)

Out[119]: (2601860978304, 2601860978304)

In [120]: mydict2 = mydict.copy()

In [121]: id(mydict2)

Out[121]: 2601860863488

In [122]: mydict['Address'] = 'Hyderabad'

In [123]: mydict

Out[123]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Address': 'Hyderabad'}

In [124]: mydict1

Out[124]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Address': 'Hyderabad'}

In [125]: mydict2

Out[125]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Address': 'Chicago'}
```

LOOP THROUGH A DICTIONARY

```
In [128]: mydict1 = {'Name':'Sidra' , 'ID': 12345 , 'DOB': 2004 , 'Address' : 'Chicago' , 'JOB' : 'Analys
          mydict1
```

```
Out[128]: {'Name': 'Sidra',
           'ID': 12345,
           'DOB': 2004,
           'Address': 'Chicago',
           'JOB': 'Analyst'}
```

```
In [129]: for i in mydict1:
              print(i , ':' , mydict1[i]) # Key & value pair
```

```
Name : Sidra
ID : 12345
DOB : 2004
Address : Chicago
JOB : Analyst
```

```
In [130]: for i in mydict1:
              print(mydict1[i]) # Dictionary items
```

```
Sidra
12345
2004
Chicago
Analyst
```

DICTIONARY MEMBERSHIP

```
In [131]: mydict1 = {'Name':'Sidra' , 'ID': 12345 , 'DOB': 2001 , 'Job': 'Analyst'}
          mydict1
```

```
Out[131]: {'Name': 'Sidra', 'ID': 12345, 'DOB': 2001, 'Job': 'Analyst'}
```

```
In [132]: 'Name' in mydict1 # Test if a key is in a dictionary or not.
```

```
Out[132]: True
```

```
In [133]: 'Sidra' in mydict1 # Membership test can be only done for keys.
```

```
Out[133]: False
```

```
In [134]: 'ID' in mydict1
```

```
Out[134]: True
```

```
In [135]: 'Address' in mydict1
```

```
Out[135]: False
```

ALL / ANY

The all() method returns:

. True - If all all keys of the dictionary are true

. False - If any key of the dictionary is false

The any() function returns True if any key of the dictionary is True. If not, any() returns False.

In [136]:

```
mydict1 = {'Name':'Sidra' , 'ID': 12345 , 'DOB': 2004 , 'Job': 'Analyst'}
mydict1
```

Out[136]:

```
{'Name': 'Sidra', 'ID': 12345, 'DOB': 2004, 'Job': 'Analyst'}
```

In [137]:

```
all(mydict1) # Will Return false as one value is false (Value 0)
```

Out[137]:

```
True
```