Date : 3rd - 09 - 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: Python Sets and Dictionary

Sets: collection of distinct objects, sets can be written as a set of comma-separated values (items) between flower brackets

Ex: { 1, 4 ,7 ,9, 11, 23, 5, 7}

```
sets.py > ...
1    a = {1,3,5,9,8,2,0}
2    print(a) # output : {0,1,2,3,5,8,9}
3
```

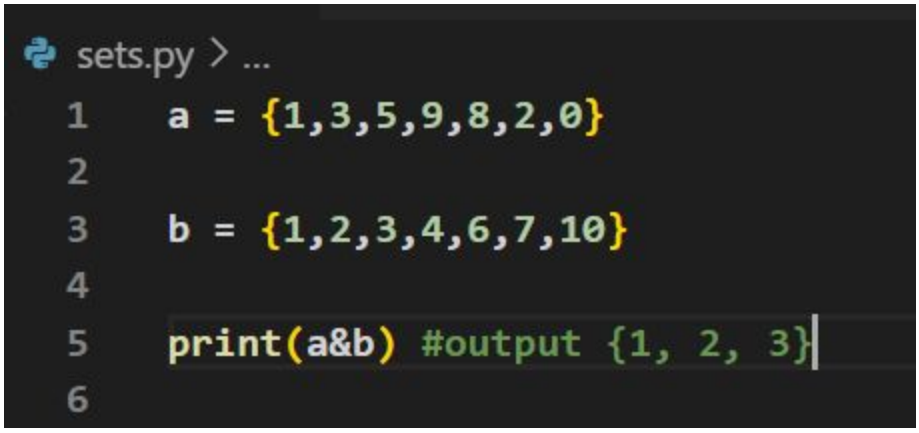
Set Operations:

1) **union (a|b or a.union(b))**

```
sets.py > ...
1    a = {1,3,5,9,8,2,0}
2
3    b = {1,2,3,4,6,7,10}
4
5    print(a|b) #output {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
6
7
```

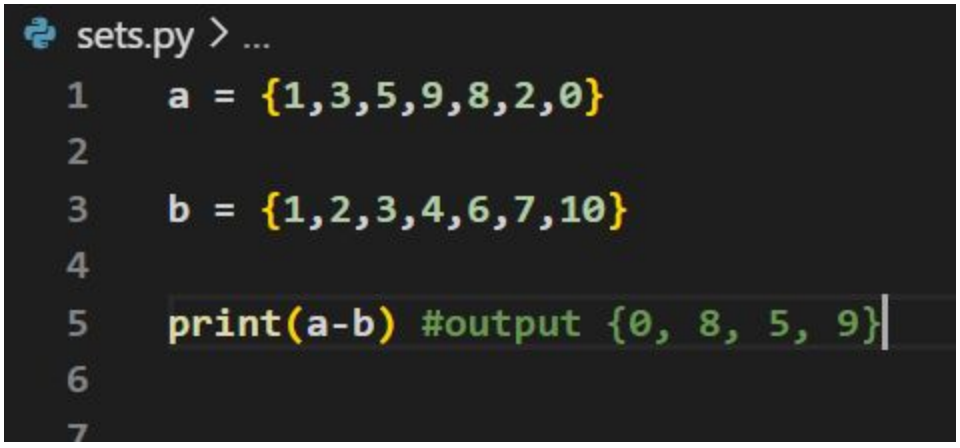
It merge two sets and will remove duplicates elements

2) **Intersection (A&b or A.Intersection(b))**



It will print only common Elements

3) **difference (a-b or A.difference(b)) :** To find difference in between sets.



What are the operations supported by seet

```
'add',
'clear',
'copy',
'difference',
'difference_update',
'discard',
'intersection',
'intersection_update',
'isdisjoint',
'issubset',
```

```
'issuperset',
'pop',
'remove',
'symmetric_difference',
'symmetric_difference_update',
'union',
'update']
```

ADD: add one element at a time

sets.py > ...

```
1   a = {1,3,5,9,8,2,0}
2
3   b = {1,2,3,4,6,7,10}
4
5   a.add(55)
6   print(a) #output {0, 1, 2, 3, 5, 8, 9, 55}
7
```

Update : we can adds more than one element

```
{1, 2, 2.5, 'abc', 'bab'}
```

```
a.add(3) #only adds one element at a time
a
```

```
{1, 2, 2.5, 3, 'abc', 'bab'}
```

```
a.update([5,7,9])  #adds more than one elem at a time
a
```

```
{1, 2, 2.5, 3, 5, 7, 9, 'abc', 'bab'}
```

Frozentset: Frozen sets in Python are immutable objects that only support methods and operators that produce a result without affecting the frozen set or sets to which they are applied. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation. If no parameters are passed, it returns an empty frozenset.

```
sets.py > ...
1   a = {1,3,5,9,8,2,0}
2
3   b = frozenset(a)
4
5   print(b) # output frozenset({0, 1, 2, 3, 5, 8, 9})
6
7   b.add(5)
8
9   print(b) #output AttributeError: 'frozenset' object has no attribute 'add'
10
```

- Sets are mutable, there is a way to immutable is called frozenset.
- Unordered.
- Unidex.
- unique/ duplicate not supported.
- To print empty set (syntax: set())

Dictionary : Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only a single value as an element, Dictionary holds **`key:value`** pair. Key value is provided in the dictionary to make it more optimized.

```
Dictionary.py > ...
1   data = {'name':'Chikki','age':"6",'class':'frist'}
2
3   print(data) # output {'name': 'Chikki', 'age': '6', 'class': 'frist'}
4
```

Herer name, age class are keys ; chikki,6,frist are values

Keys must be unique/ value whether it is unique doesn't matter.

Dictionary.py > ...

```python
data = {'name':'Chikki','age':"6",'class':'frist'}

print(data) # output {'name': 'Chikki', 'age': '6', 'class': 'frist'}

print(data['name']) #output Chikki

print(data['place']) #output KeyError: 'place'

```

It u try to access out of element it will give error

Dictionary supported operations:

```
'clear',
'copy',
'fromkeys',
'get',
'items',
'keys',
'pop',
'popitem',
'setdefault',
'update',
'values']
```

Merging Two list

Dictionary.py ×

Dictionary.py > ...

```python
data = ['name','age','class']
data1 = ['chikki','6','frist']
details = dict(zip(data,data1))
print(details) #output {'name': 'chikki', 'age': '6', 'class': 'frist'}

```

Update:

```python
data = {'name': 'chikki', 'age': '6', 'class': 'frist'}
print(data) #output {'name': 'chikki', 'age': '6', 'class': 'frist'}

#update
data['school'] = 'army public school'

print(data) #output {'name': 'chikki', 'age': '6', 'class': 'frist', 'school': 'army public school'}

```

Delete :

```python
data = {'name': 'chikki', 'age': '6', 'class': 'frist'}
print(data) #output {'name': 'chikki', 'age': '6', 'class': 'frist'}

#update
data['school'] = 'army public school'

print(data) #output {'name': 'chikki', 'age': '6', 'class': 'frist', 'school': 'army public school'}

del data['school']

print(data) #output {'name': 'chikki', 'age': '6', 'class': 'frist'}

```

Nested dictionary:

```python
#nested dictionary
data = {'name': 'chikki', 'age': '6', 'class': 'frist', 'school' : {'branch': 'sainikpuri', 'place':'hyderabad'}}
print(data) #o/p {'name': 'chikki', 'age': '6', 'class': 'frist', 'school': {'branch': 'sainikpuri', 'place': 'hyderabad'}}


```

To find key and values:

```python
data = {'name': 'chikki', 'age': '6', 'class': 'frist'}
print(data) #o/p {'name': 'chikki', 'age': '6', 'class': 'frist'}
print(data.keys()) #o/p dict_keys(['name', 'age', 'class'])
print(data.values()) #o/p  dict_values(['chikki', '6', 'frist'])


```

To access nested dictionary :

```python
#nested dictionary
data = {'name': 'chikki', 'age': '6', 'class': 'frist', 'school' : {'branch': 'sainikpuri', 'place':'hyderabad'}}
print(data['school']['branch']) #sainikpuri
print(data['school']['place']) #hyderabad
```

MCQ's:

Point: 75%

How to declare empty Dict — 2

- a = {}
- a = dict()
- ✓ Both — 42.86%

How to declare empty Set — 2

- ✓ a = set() — 85.71%
- a = {}

Dict are unordered — 2

- ✓ True — 80.95%
- False

Sets are Ordered — 2

- True
- ✓ False — 80.95%