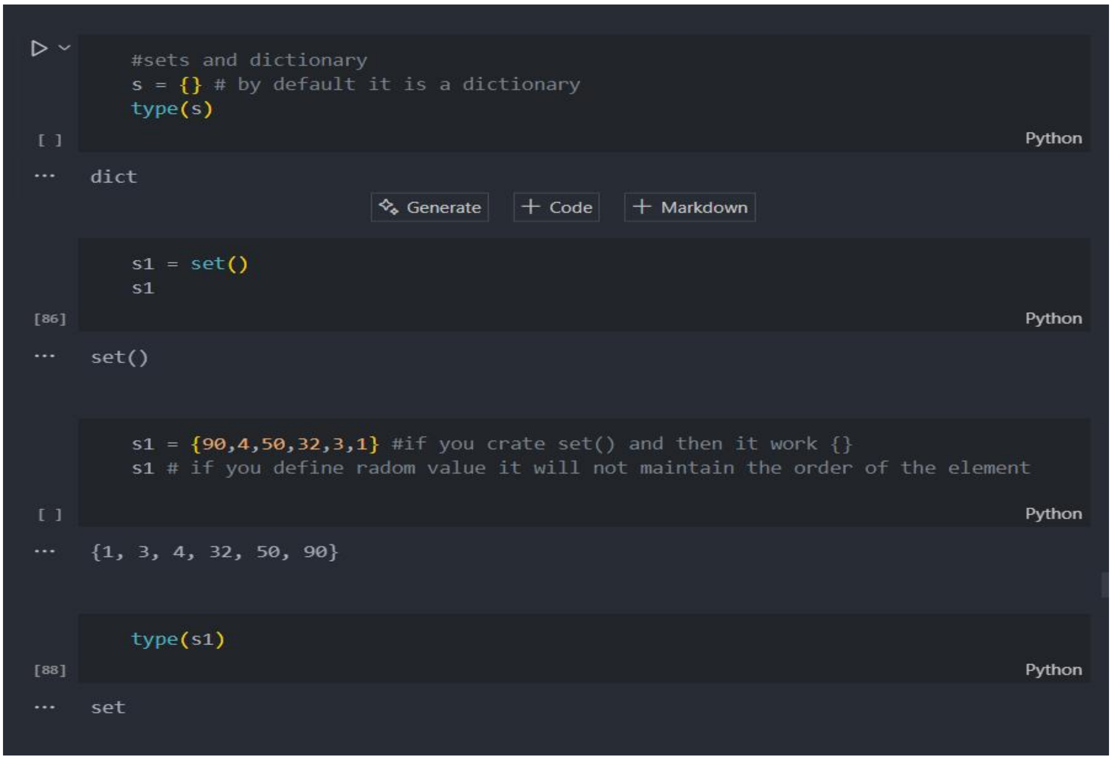
Sets



```
#sets and dictionary
s = {} # by default it is a dictionary
type(s)
```

```
dict
```

```
s1 = set()
s1
```

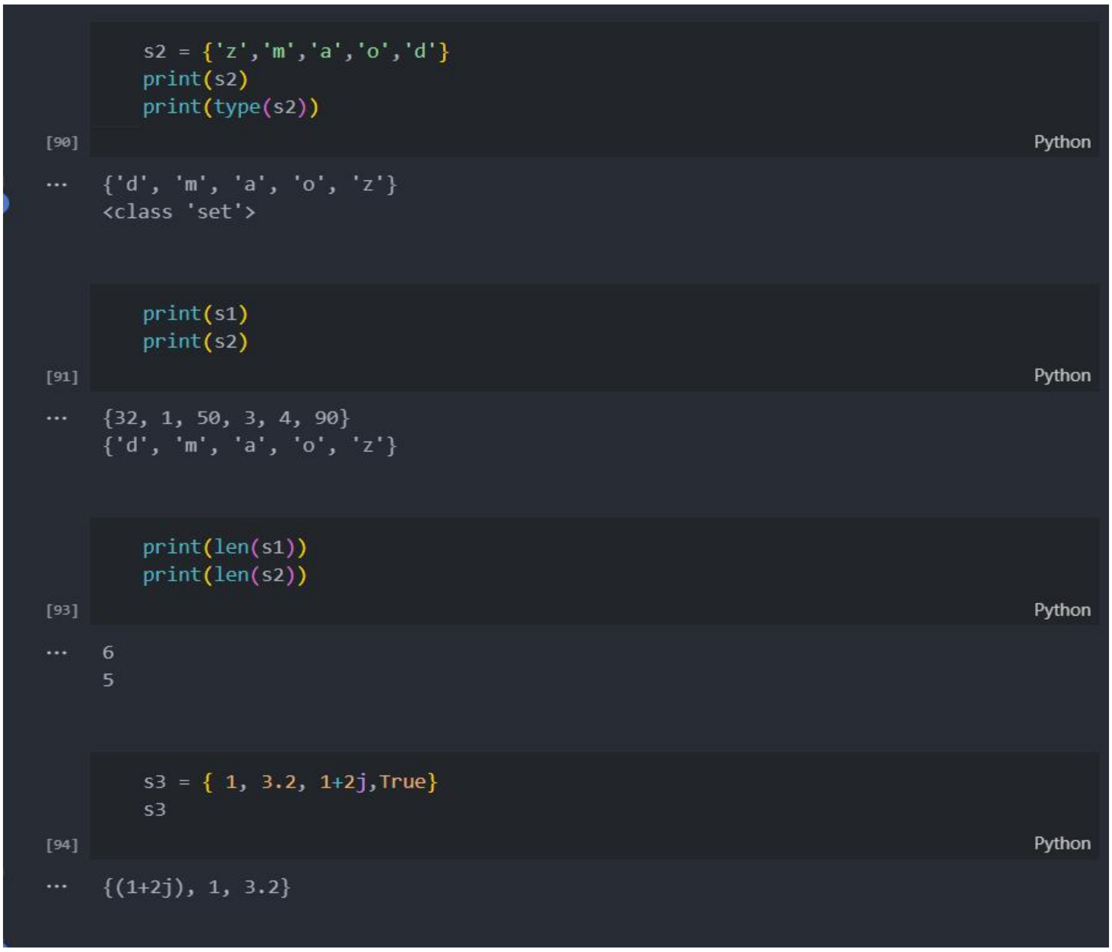
```
set()
```

```
s1 = {90,4,50,32,3,1} #if you crate set() and then it work {}
s1 # if you define radom value it will not maintain the order of the element
```

```
{1, 3, 4, 32, 50, 90}
```

```
type(s1)
```

```
set
```



```
s2 = {'z','m','a','o','d'}
print(s2)
print(type(s2))
```

```
{'d', 'm', 'a', 'o', 'z'}
<class 'set'>
```

```
print(s1)
print(s2)
```

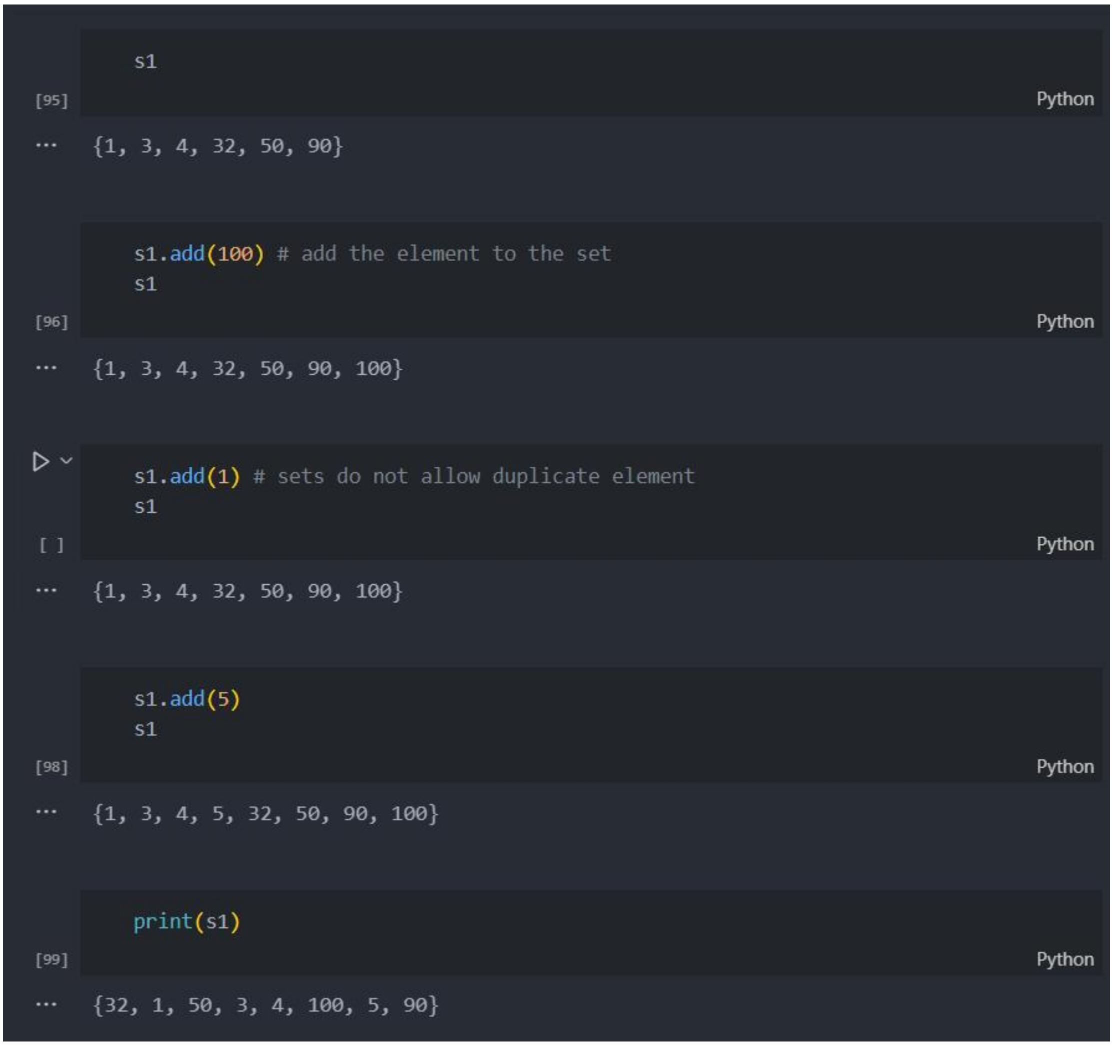
```
{32, 1, 50, 3, 4, 90}
{'d', 'm', 'a', 'o', 'z'}
```

```
print(len(s1))
print(len(s2))
```

```
6
5
```

```
s3 = { 1, 3.2, 1+2j,True}
s3
```

```
{(1+2j), 1, 3.2}
```



```
s1
```

```
{1, 3, 4, 32, 50, 90}
```

```
s1.add(100) # add the element to the set
s1
```

```
{1, 3, 4, 32, 50, 90, 100}
```

```
s1.add(1) # sets do not allow duplicate element
s1
```

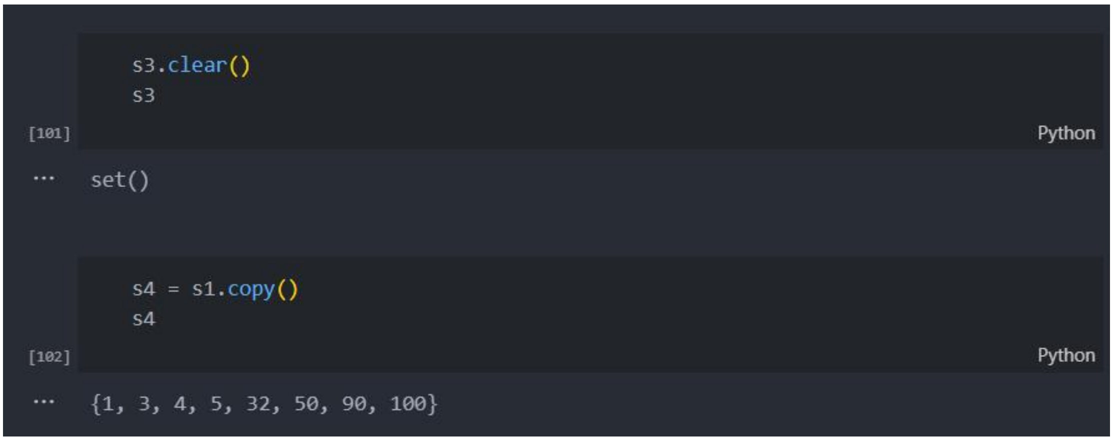
```
{1, 3, 4, 32, 50, 90, 100}
```

```
s1.add(5)
s1
```

```
{1, 3, 4, 5, 32, 50, 90, 100}
```

```
print(s1)
```

```
{32, 1, 50, 3, 4, 100, 5, 90}
```

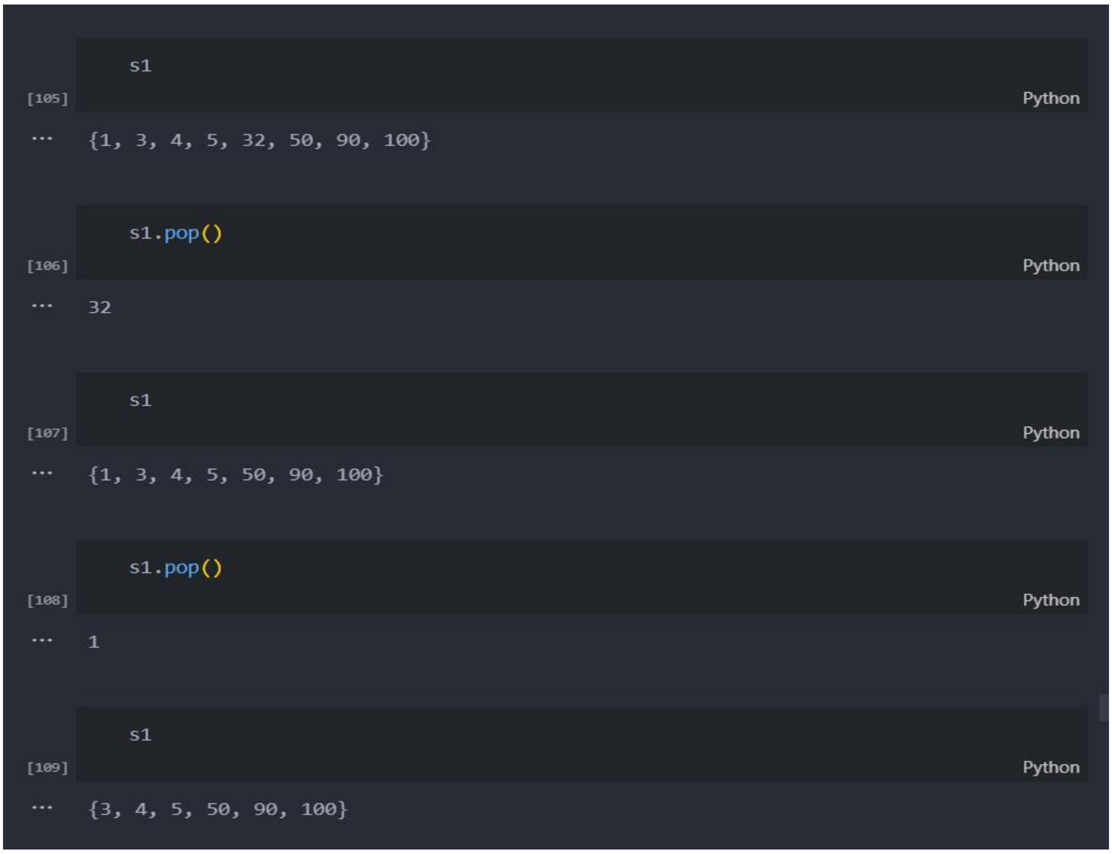


```
s3.clear()
s3
```

```
set()
```

```
s4 = s1.copy()
s4
```

```
{1, 3, 4, 5, 32, 50, 90, 100}
```



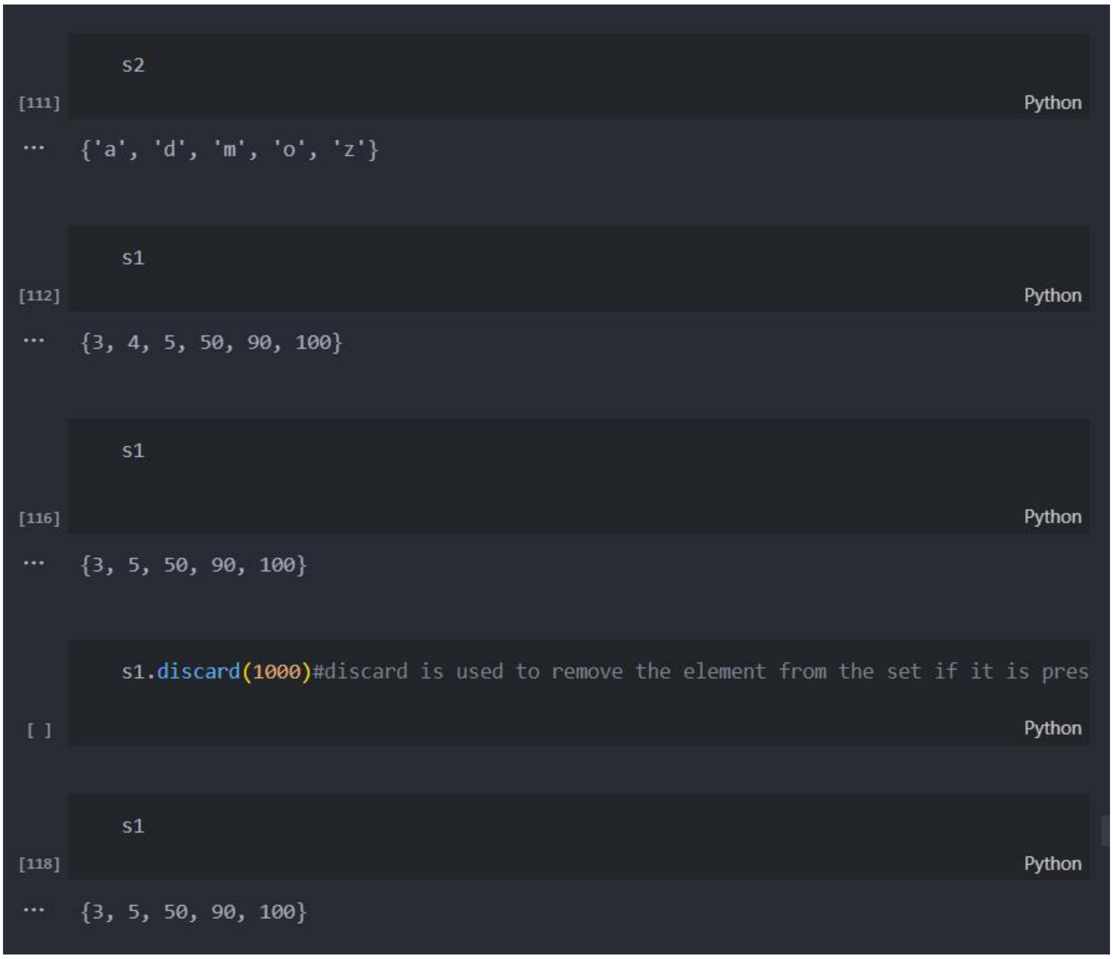
```
[105]   s1
... {1, 3, 4, 5, 32, 50, 90, 100}

[106]   s1.pop()
... 32

[107]   s1
... {1, 3, 4, 5, 50, 90, 100}

[108]   s1.pop()
... 1

[109]   s1
... {3, 4, 5, 50, 90, 100}
```



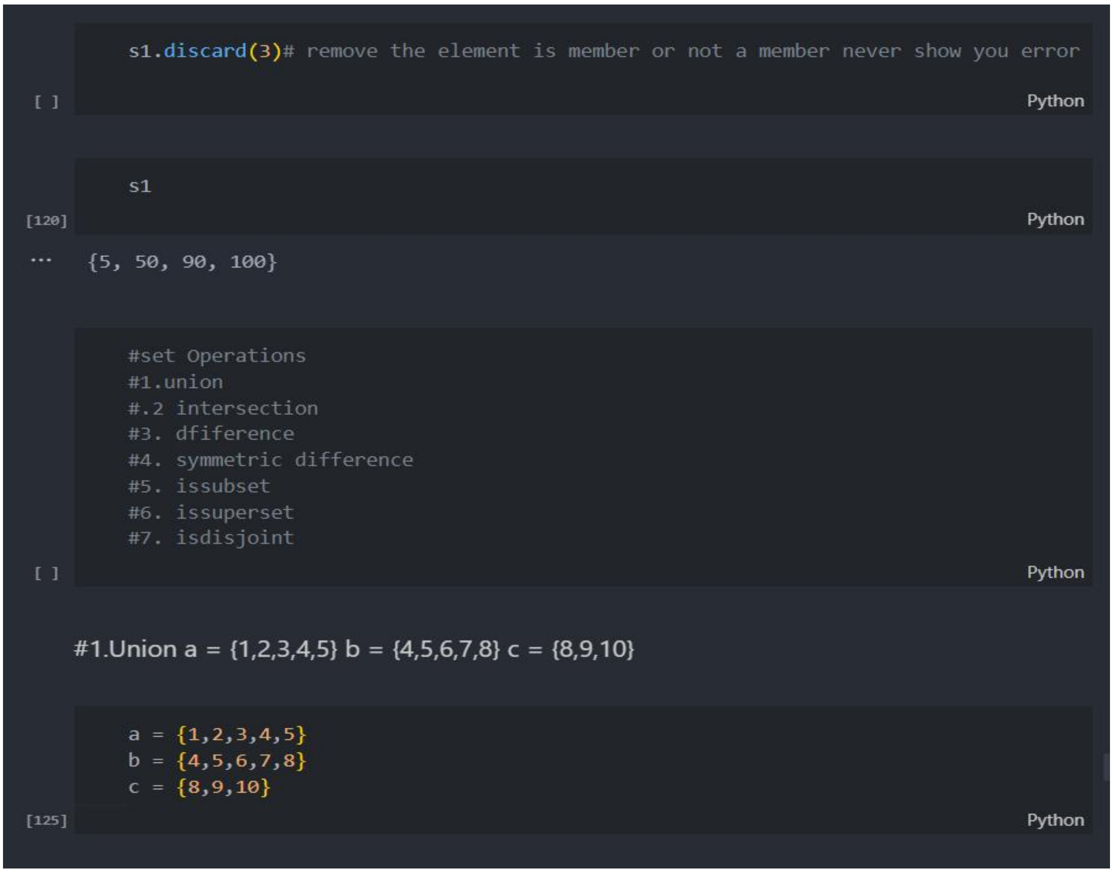
```
[111]   s2
... {'a', 'd', 'm', 'o', 'z'}

[112]   s1
... {3, 4, 5, 50, 90, 100}

[116]   s1
... {3, 5, 50, 90, 100}

[ ]     s1.discard(1000)#discard is used to remove the element from the set if it is pres

[118]   s1
... {3, 5, 50, 90, 100}
```



```
[ ]   s1.discard(3)# remove the element is member or not a member never show you error
```

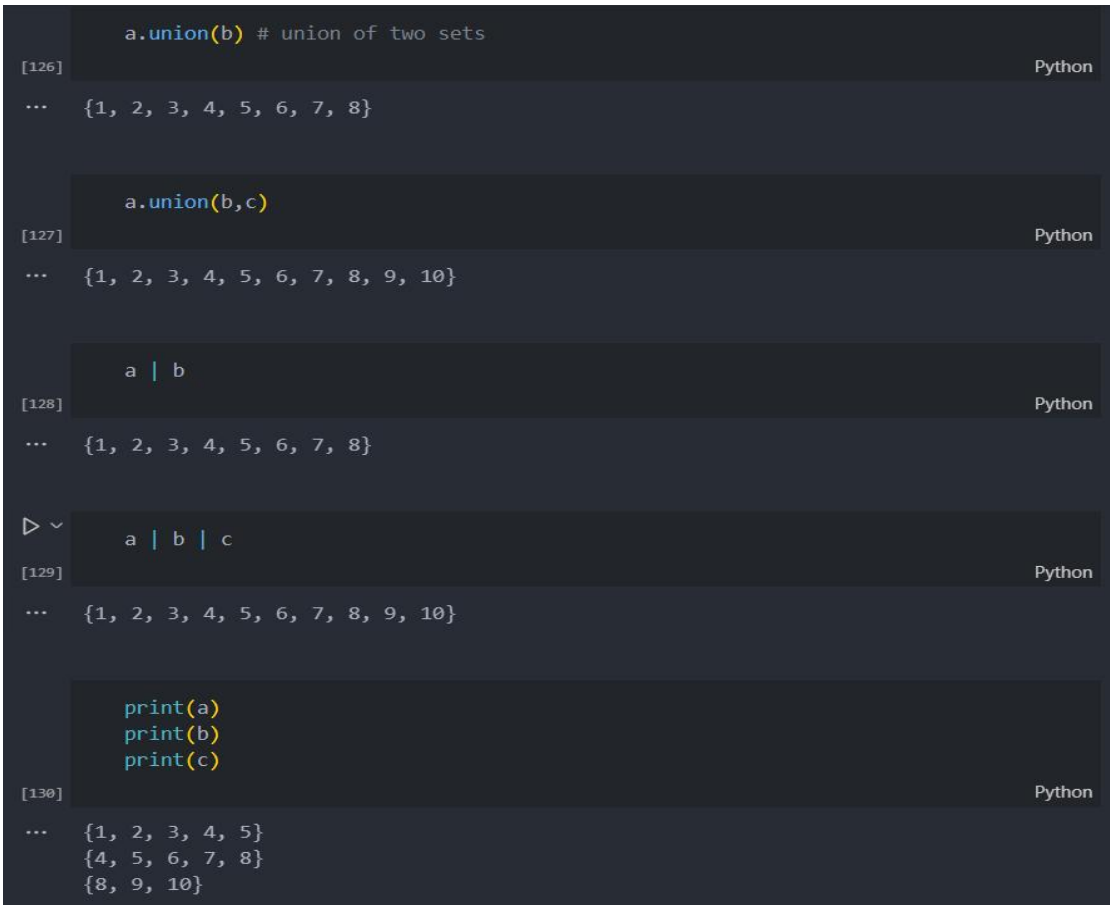
```
[120] s1
```

```
{5, 50, 90, 100}
```

```
[ ]   #set Operations
      #1.union
      #.2 intersection
      #3. dfiference
      #4. symmetric difference
      #5. issubset
      #6. issuperset
      #7. isdisjoint
```

#1.Union a = {1,2,3,4,5} b = {4,5,6,7,8} c = {8,9,10}

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



```
[126] a.union(b) # union of two sets
```

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

```
[127] a.union(b,c)
```

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

```
[128] a | b
```

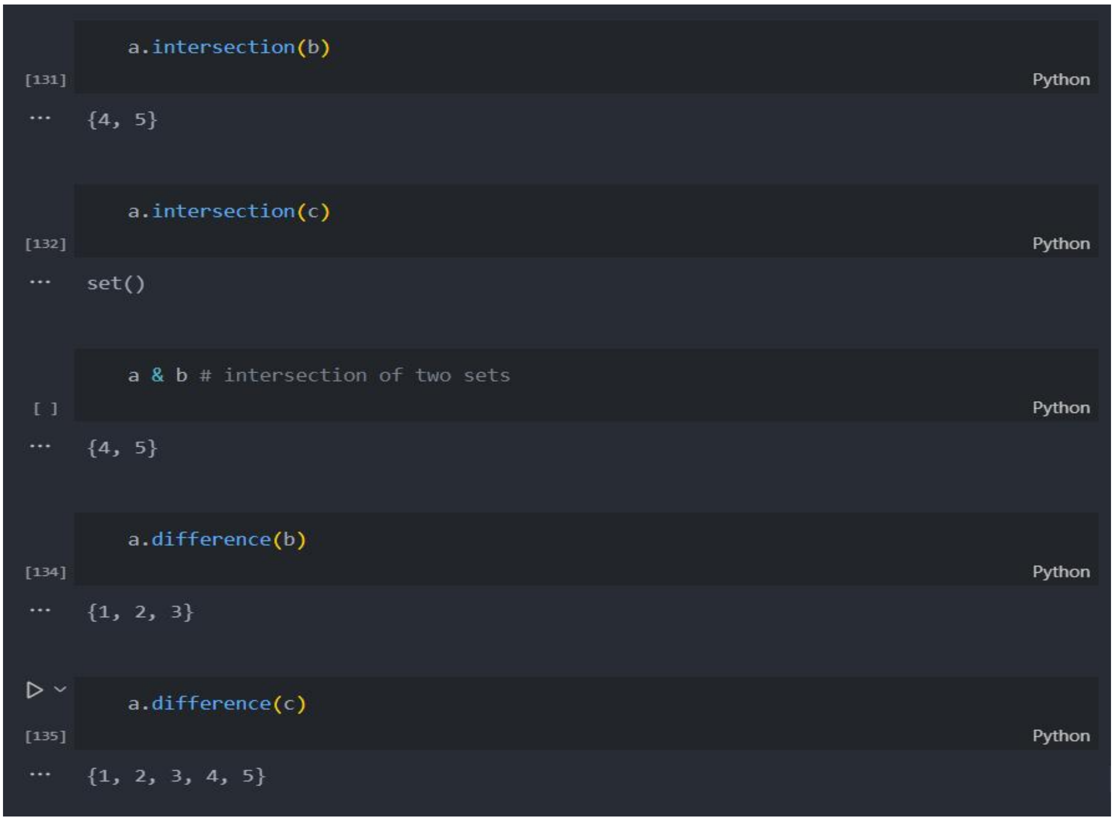
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
[129] a | b | c
```

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

```
[130] print(a)
      print(b)
      print(c)
```

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



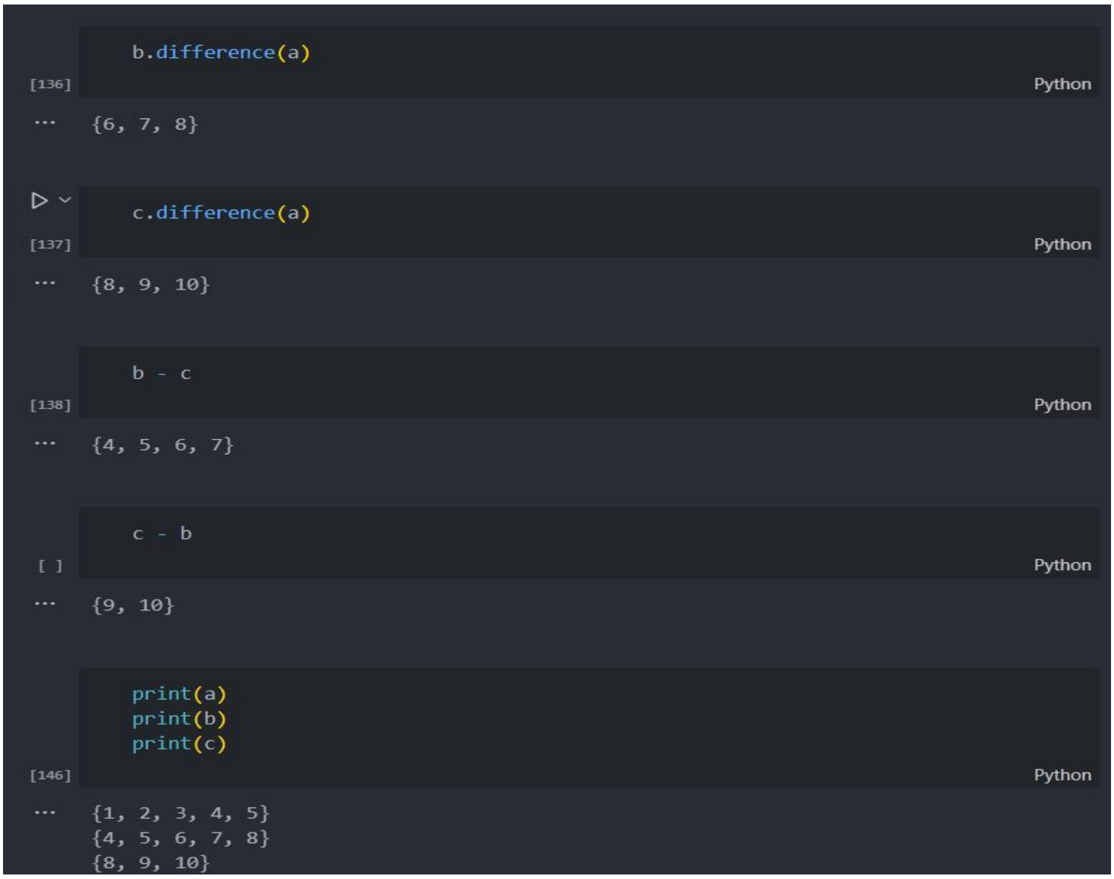
```
[131]  a.intersection(b)
...    {4, 5}

[132]  a.intersection(c)
...    set()

[ ]    a & b # intersection of two sets
...    {4, 5}

[134]  a.difference(b)
...    {1, 2, 3}

[135]  a.difference(c)
...    {1, 2, 3, 4, 5}
```



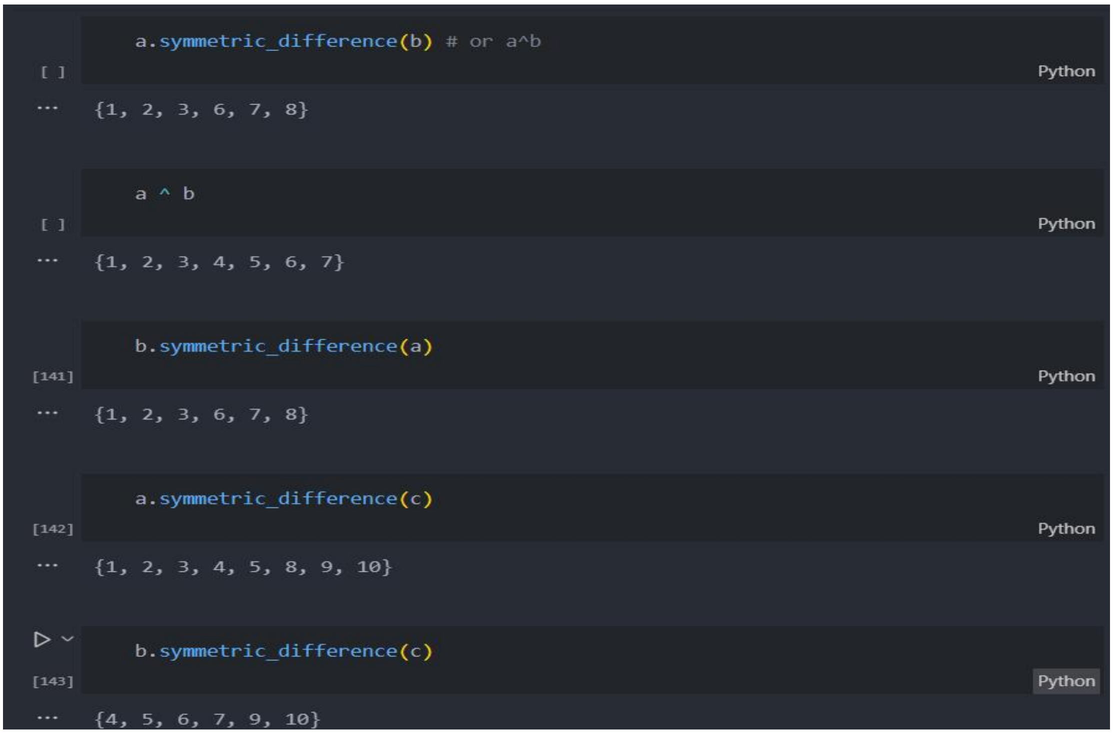
```
[136]  b.difference(a)
...    {6, 7, 8}

[137]  c.difference(a)
...    {8, 9, 10}

[138]  b - c
...    {4, 5, 6, 7}

[ ]    c - b
...    {9, 10}

[146]  print(a)
       print(b)
       print(c)
...    {1, 2, 3, 4, 5}
       {4, 5, 6, 7, 8}
       {8, 9, 10}
```



```python
a.symmetric_difference(b) # or a^b
```

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

```python
a ^ b
```

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

```python
b.symmetric_difference(a)
```

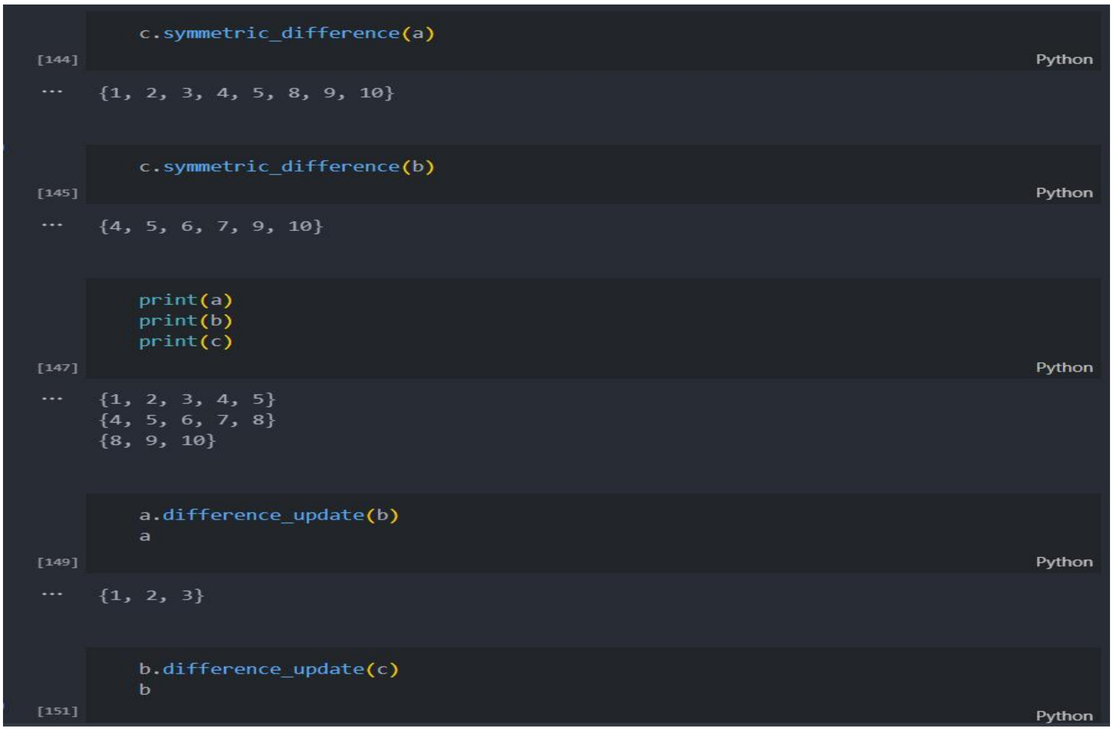
```
{1, 2, 3, 6, 7, 8}
```

```python
a.symmetric_difference(c)
```

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

```python
b.symmetric_difference(c)
```

```
{4, 5, 6, 7, 9, 10}
```



```python
c.symmetric_difference(a)
```

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

```python
c.symmetric_difference(b)
```

```
{4, 5, 6, 7, 9, 10}
```

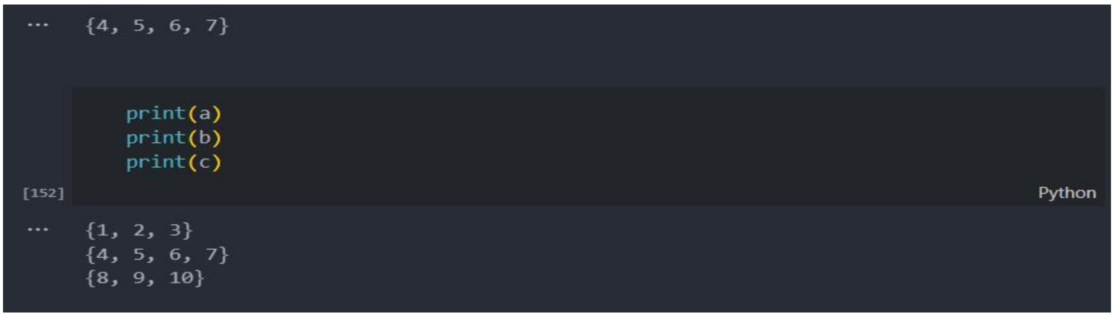
```python
print(a)
print(b)
print(c)
```

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

```python
a.difference_update(b)
a
```

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

```python
b.difference_update(c)
b
```



```
{4, 5, 6, 7}
```

```python
print(a)
print(b)
print(c)
```

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