



4.symmetric_difference():

x.symmetric_difference(y) or x^y

Returns elements present in either x or y but not in both

Eg:

```
x={10,20,30,40}
y={30,40,50,60}
print(x.symmetric_difference(y))    #{10, 50, 20, 60}
print(x^y)                          #{10, 50, 20, 60}
```

Membership operators: (in , not in)

Eg:

```
1. s=set("durga")
2. print(s)
3. print('d' in s)
4. print('z' in s)
5.
6. Output
7. {'u', 'g', 'r', 'd', 'a'}
8. True
9. False
```

Set Comprehension:

Set comprehension is possible.

```
s={x*x for x in range(5)}
print(s) #{0, 1, 4, 9, 16}
```

```
s={2**x for x in range(2,10,2)}
print(s)    #{16, 256, 64, 4}
```

set objects won't support indexing and slicing:

Eg:

```
s={10,20,30,40}
print(s[0])    ==>TypeError: 'set' object does not support indexing
print(s[1:3]) ==>TypeError: 'set' object is not subscriptable
```