sets

```
In [1]: s={}
        s
```

```
Out[1]: {}
```

```
In [2]: type(s)
```

```
Out[2]: dict
```

```
In [5]: s1=set()#empty set
        s1
```

```
Out[5]: set()
```

```
In [4]: type(s1)
```

```
Out[4]: set
```

```
In [6]: s1.add(70)
```

```
In [7]: s1
```

```
Out[7]: {70}
```

```
In [10]: s1.add(40)# herer user shd not declare more than one argument so here we only gi
         s1.add(56)
         s1.add(24)
```

```
In [11]: s1
```

```
Out[11]: {24, 40, 56, 70}
```

```
In [12]: s1.add(24)
```

```
In [13]: s1
```

```
Out[13]: {24, 40, 56, 70}
```

```
In [14]: s1[0]# indexing is not allowed in sets
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 s1[0]

TypeError: 'set' object is not subscriptable
```

```
In [15]: s1[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 s1[:]

TypeError: 'set' object is not subscriptable
```

In [16]:
```
s1
```
Out[16]:
```
{24, 40, 56, 70}
```

In [17]:
```
s2=set()
```

In [20]:
```
s2.add(10)
s2.add(1.2)
s2.add(True)
```

In [24]:
```
s2
```
Out[24]:
```
{1.2, 10, True, 'fsds'}
```

In [22]:
```
s2.add('fsds')
```

In [23]:
```
s2
```
Out[23]:
```
{1.2, 10, True, 'fsds'}
```

In [25]:
```
print(s1)
print(s2)
```
```
{40, 24, 56, 70}
{1.2, 10, True, 'fsds'}
```

In [26]:
```
s1
```
Out[26]:
```
{24, 40, 56, 70}
```

In [27]:
```
s1
s2
```
Out[27]:
```
{1.2, 10, True, 'fsds'}
```

In [28]:
```
id(s1)==id(s2)
```
Out[28]:
```
False
```

In [31]:
```
s3=s2.copy()
```

In [32]:
```
s3
```
Out[32]:
```
{1.2, 10, True, 'fsds'}
```

In [33]:
```
s2.pop()
```
Out[33]:
```
1.2
```

In [34]:
```
s2
```
Out[34]:
```
{10, True, 'fsds'}
```

In [35]:
```
s3
```
Out[35]:
```
{1.2, 10, True, 'fsds'}
```

In [36]:
```
s3.remove(10)
```

In [37]:
```
s3
```
Out[37]:
```
{1.2, True, 'fsds'}
```

In [38]:
```
s3.remove(1000)
```
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[38], line 1
----> 1 s3.remove(1000)

KeyError: 1000
```

In [ ]:
```
# in sets if we use remove and there is no element it gives error but unlike rem
#discard never gives error as an result it just prints the whole set as an answe
# it also doesnot give a bug also
```

In [39]:
```
s3.discard(1000)
s3
```
Out[39]:
```
{1.2, True, 'fsds'}
```

In [40]:
```
for i in s1:
    print(i)
```
```
40
24
56
70
```

In [42]:
```
for i in enumerate(s1):
    print(i)
```
```
(0, 40)
(1, 24)
(2, 56)
(3, 70)
```

In [ ]:
```
#set operations
```

In [43]:
```
a={1,2,3,4,5}
b={4,5,6,7,8}
c={8,9,10}
```

In [44]:
```
a.union(b)
```
Out[44]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [45]: a|c

Out[45]: {1, 2, 3, 4, 5, 8, 9, 10}

In [46]: b|c

Out[46]: {4, 5, 6, 7, 8, 9, 10}

In [47]: a|b|c

Out[47]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [48]: print(a)
         print(b)
         print(c)

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

In [49]: a.difference(b)

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

In [50]: a.difference(c)

Out[50]: {1, 2, 3, 4, 5}

In [51]: b.difference(c)

Out[51]: {4, 5, 6, 7}

In [52]: c.difference(b)

Out[52]: {9, 10}

In [54]: c.difference(a)

Out[54]: {8, 9, 10}

In [55]: c.difference(c)

Out[55]: set()

In [ ]:
```