In [1]:
```
s={}
type(s)
```

Out[1]:
```
dict
```

In [2]:
```
s1=set()
s1
```

Out[2]:
```
set()
```

In [3]:
```
s1={90,4,50,32,3,1}
s1
```

Out[3]:
```
{1, 3, 4, 32, 50, 90}
```

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

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

In [5]:
```
s2={'z','m','a','o','d'}
s2
```

Out[5]:
```
{'a', 'd', 'm', 'o', 'z'}
```

In [6]:
```
type(s2)
```

Out[6]:
```
set
```

In [7]:
```
print(s1)
print(s2)
```

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

In [8]:
```
s2
```

Out[8]:
```
{'a', 'd', 'm', 'o', 'z'}
```

In [10]:
```
s3={1,3.2,'nit',1+2j,False}
s3
```

Out[10]:
```
{(1+2j), 1, 3.2, False, 'nit'}
```

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

Out[11]:
```
{1, 3, 4, 32, 50, 90}
```

In [12]:
```
s1.add(1)
```

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

```
Out[13]:  {1, 3, 4, 32, 50, 90}

In [14]:  s1.add(100)
          s1

Out[14]:  {1, 3, 4, 32, 50, 90, 100}

In [15]:  s1.add(5)
          s1

Out[15]:  {1, 3, 4, 5, 32, 50, 90, 100}

In [16]:  print(s1)

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

In [17]:  s3

Out[17]:  {(1+2j), 1, 3.2, False, 'nit'}

In [18]:  s3.clear()

In [19]:  s1

Out[19]:  {1, 3, 4, 5, 32, 50, 90, 100}

In [20]:  s4=s1.copy()
          s4

Out[20]:  {1, 3, 4, 5, 32, 50, 90, 100}

In [21]:  s1

Out[21]:  {1, 3, 4, 5, 32, 50, 90, 100}

In [22]:  s1[0]

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         Cell In[22], line 1
         ----> 1 s1[0]

         TypeError: 'set' object is not subscriptable

In [23]:  s1

Out[23]:  {1, 3, 4, 5, 32, 50, 90, 100}

In [24]:  s1[1:5]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[24], line 1
----> 1 s1[1:5]

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

```
In [25]: s1
```

```
Out[25]: {1, 3, 4, 5, 32, 50, 90, 100}
```

```
In [26]: s1.pop()
```

```
Out[26]: 32
```

```
In [27]: s1
```

```
Out[27]: {1, 3, 4, 5, 50, 90, 100}
```

```
In [28]: s1.pop()
```

```
Out[28]: 1
```

```
In [29]: s1
```

```
Out[29]: {3, 4, 5, 50, 90, 100}
```

```
In [30]: s1.pop(0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[30], line 1
----> 1 s1.pop(0)

TypeError: set.pop() takes no arguments (1 given)
```

```
In [31]: s1
```

```
Out[31]: {3, 4, 5, 50, 90, 100}
```

```
In [32]: s1.remove(1000)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[32], line 1
----> 1 s1.remove(1000)

KeyError: 1000
```

```
In [33]: s1
```

```
Out[33]: {3, 4, 5, 50, 90, 100}
```

```
In [34]: s1.discard(3)
         s1
```

```
Out[34]: {4, 5, 50, 90, 100}
```

```
In [35]: s1
```

```
Out[35]: {4, 5, 50, 90, 100}
```

```
In [36]: s1.discard(3)
         s1
```

```
Out[36]: {4, 5, 50, 90, 100}
```

```
In [37]: 1000 in s1
```

```
Out[37]: False
```

```
In [38]: s1.add(1000)
         s1
```

```
Out[38]: {4, 5, 50, 90, 100, 1000}
```

Set Operation

```
In [39]: a={1,2,3,4,5}
         b={4,5,6,7,8}
         c={8,9,10}
```

```
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 [43]: a|b
```

```
Out[43]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [44]: a|b|c
```

```
Out[44]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [45]: print(a)
         print(b)
         print(c)
```

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

```
In [46]: a.intersection(b)

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

```
In [48]: a.intersection(c)

Out[48]: set()
```

```
In [49]: a&b

Out[49]: {4, 5}
```

```
In [50]: print(a)
         print(b)
         print(c)

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

```
In [52]: a.difference(b)

Out[52]: {1, 2, 3}
```

```
In [53]: b.difference(a)

Out[53]: {6, 7, 8}
```

```
In [54]: b-c

Out[54]: {4, 5, 6, 7}
```

```
In [55]: c-b

Out[55]: {9, 10}
```

```
In [56]: a-c

Out[56]: {1, 2, 3, 4, 5}
```

```
In [57]: print(a)
         print(b)
         print(c)

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

```
In [59]: b.difference_update(c)
         b

Out[59]: {4, 5, 6, 7}
```

In [61]:

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

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

In [62]:

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

Out[62]:

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

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: