set

In [2]:
```
# creating a empty set
s=set()
```

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

Out[4]: set

In [6]:
```
# set accepts all data types
s={1,2.3,"hello",1+2j,True}
s
```

Out[6]: {(1+2j), 1, 2.3, 'hello'}

In [8]:
```
# length of the set
len(s)
```

Out[8]: 4

In [10]:
```
#loop in set
for i in s:
    print(i)
```

```
1
2.3
hello
(1+2j)
```

In [16]:
```
for i in enumerate (s):
    print(i)
```

```
(0, 1)
(1, 2.3)
(2, 'hello')
(3, (1+2j))
```

In [18]:
```
# set membership
1 in s
```

Out[18]: True

In [20]:
```
2.3 in s
```

Out[20]: True

In [28]:
```
if 2.3 in s:
    print("2.3 belongs to the set s")
else:
    print("2.3 do not belongs to s")
```

```
2.3 belongs to the set s
```

set functions

In [31]:
```
s.add("hello")
s
```

Out[31]:
```
{(1+2j), 1, 2.3, 'hello'}
```

In [21]:
```
s1={"hii",2.3,25,True,"hello"}
s1
```

Out[21]:
```
{2.3, 25, True, 'hello', 'hii'}
```

In [73]:
```
s2=s1
s2
```

Out[73]:
```
{2.3, 25, True, 'hello', 'hii'}
```

In [13]:
```
s3={2,"hii",True,2.5,"hello"}
s3
```

Out[13]:
```
{2, 2.5, True, 'hello', 'hii'}
```

In [23]:
```
s1.clear()
s1
```

Out[23]:
```
set()
```

In [25]:
```
s3.discard(2)
s3
```

Out[25]:
```
{2.5, True, 'hello', 'hii'}
```

In [27]:
```
s3.remove(2.5)
```

In [29]:
```
s3
```

Out[29]:
```
{True, 'hello', 'hii'}
```

In [31]:
```
s3.pop()
```

Out[31]:
```
True
```

In [35]:
```
s4={4,6 ,"hello"}
```

set operations

In [41]:
```
s3.union(s4)
```

Out[41]: {4, 6, 'hello', 'hii'}

In [43]:
```
s3.intersection(s4)
```

Out[43]: {'hello'}

In [2]:
```
s4={2,3,4,5,6}
s5={5,6,7,8,9}
```

In [4]:
```
s4.difference(s5)
```

Out[4]: {2, 3, 4}

In [10]:
```
s5&s4 # difference can be denoted with "&" symbol
```

Out[10]: {5, 6}

In [14]:
```
s5|s4 # union can be denoted with "|" symbol
```

Out[14]: {2, 3, 4, 5, 6, 7, 8, 9}

In [20]:
```
s5.difference_update(s4)
```

In [22]:
```
s4
```

Out[22]: {2, 3, 4, 5, 6}

In [26]:
```
s6={2,3,4,5,6}
s7={5,6,7,8,9}
```

In [28]:
```
s6.symmetric_difference(s7)
```

Out[28]: {2, 3, 4, 7, 8, 9}

In [48]:
```
my_set1={1,2,3,4,5,6,7}
my_set2={3,4,5,6,7}
my_set3={9,0,10,20,30}
```

In [50]:
```
my_set2.issuperset(my_set1)
```

Out[50]: False

In [52]:
```
my_set1.issuperset(my_set2)
```

Out[52]: True

In [54]:
```
my_set2.issubset(my_set1)
```

Out[54]: True

In [58]:
```
my_set2.isdisjoint(my_set3)
```

Out[58]: True

other builtin functions

In [61]:
```
len(my_set1)
```

Out[61]: 7

In [63]:
```
max(my_set1)
```

Out[63]: 7

In [65]:
```
min(my_set1)
```

Out[65]: 1

In [67]:
```
list(enumerate(my_set1))
```

Out[67]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7)]

In [75]:
```
d=sorted(my_set1)
```

In [91]:
```
d
```

Out[91]: [7, 6, 5, 4, 3, 2, 1]

In [93]:
```
d=sorted(my_set1,reverse=True)
d
```

Out[93]: [7, 6, 5, 4, 3, 2, 1]

dict

creating the dict

In [97]:
```
my_dict={}
```

In [99]:
```
type(my_dict)
```

Out[99]: dict

dict functions

In [113…
```
my_dict={1:"name",2:"age",3:"address",4:"phno"} #dict stores items as key value pai
```

```
In [113…  my_dict
```

```
Out[113…  {1: 'name', 2: 'age', 3: 'address', 4: 'phno'}
```

```
In [119…  my_dict[3]#it is not indexing we can call the vlaue by defining it key
```

```
Out[119…  'address'
```

```
In [121…  my_dict.keys()
```

```
Out[121…  dict_keys([1, 2, 3, 4])
```

```
In [123…  my_dict.values()
```

```
Out[123…  dict_values(['name', 'age', 'address', 'phno'])
```

```
In [125…  my_dict.items()
```

```
Out[125…  dict_items([(1, 'name'), (2, 'age'), (3, 'address'), (4, 'phno')])
```

```
In [129…  my_dict.pop( 4)
```

```
Out[129…  'phno'
```

```
In [133…  my_dict.popitem()    # popitem means it removes random item from the dict
```

```
Out[133…  (2, 'age')
```

dict membership

```
In [136…  dict2={1:"hii",2:"hello",3:"how r u"}
```

```
In [142…  "hii" in dict2
```

```
Out[142…  False
```

loop through dict

```
In [147…  for i in dict2:
              print(i)
```

```
1
2
3
```

add,remove & change

In [17]:
```
dic={1:"raj",2:"22",3:"csd",4:"hyd"}
```

In [19]:
```
dic[1]="ram"
```

In [21]:
```
dic
```

Out[21]:
```
{1: 'ram', 2: '22', 3: 'csd', 4: 'hyd'}
```

In [23]:
```
dic.pop(4)
```

Out[23]:
```
'hyd'
```

In [25]:
```
dic
```

Out[25]:
```
{1: 'ram', 2: '22', 3: 'csd'}
```

In [27]:
```
dic.clear()
```

In [29]:
```
dict
```

Out[29]:
```
dict
```

In [ ]:
```
dict.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: