Set

In [1]:
```
#create an empty set
myset={}
```

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

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

In [3]:
```
myset=()
type(myset)
```

Out[3]:
```
tuple
```

In [4]:
```
myset=[]
type(myset)
```

Out[4]:
```
list
```

In [5]:
```
myset=set()
type(myset)
```

Out[5]:
```
set
```

In [6]:
```
myset
```

Out[6]:
```
set()
```

In [8]:
```
print(type(myset))
```

```
<class 'set'>
```

In [9]:
```
s1=(78,-90.78,False,"String","T")
s1
```

Out[9]:
```
(78, -90.78, False, 'String', 'T')
```

In [10]:
```
print(s1)
```

```
(78, -90.78, False, 'String', 'T')
```

In [11]:
```
s2={1,1,1,2,1,3,2,1,3,4}# Set does not allow duplicate values
s2
```

Out[11]:
```
{1, 2, 3, 4}
```

In [13]:
```
list3=[1,1,1,2,1,3,2,1,3,4]# List allows duplicate values
list3
```

Out[13]:
```
[1, 1, 1, 2, 1, 3, 2, 1, 3, 4]
```

In [15]:
```
len(list3)
```

Out[15]:
```
10
```

In [16]:
```
print(dir(set),end=" ")
```

```
['__and__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__iand__', '__init__', '__init_subclass__', '__ior__', '__isub__', '__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__xor__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'intersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union', 'update']
```

In [17]:
```
s2
```

Out[17]:
```
{1, 2, 3, 4}
```

In [18]:
```
A={1,3,4,5,6,9}
B={2,4,6,8}
```

In [19]:
```
#Union
A.union(B)
```

Out[19]:
```
{1, 2, 3, 4, 5, 6, 8, 9}
```

In [20]:
```
A|B
```

Out[20]:
```
{1, 2, 3, 4, 5, 6, 8, 9}
```

In [21]:
```
#Intersection
A&B
```

Out[21]:
```
{4, 6}
```

In [22]:
```
A.intersection(B)
```

Out[22]:
```
{4, 6}
```

In [23]:
```
A-B
```

Out[23]:
```
{1, 3, 5, 9}
```

In [24]:
```
B-A
```

Out[24]:
```
{2, 8}
```

In [25]:
```
s2.pop()
```

Out[25]:
```
1
```

In [26]:
```
s2
```

Out[26]:
```
{2, 3, 4}
```

In [29]:
```
s2[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13880\2825080482.py in <module>
----> 1 s2[1]

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

In [30]:
```
# Indexing is not allowed in set because it is unorederd
```

In [31]:
```
s2[1]=23
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13880\4254879085.py in <module>
----> 1 s2[1]=23

TypeError: 'set' object does not support item assignment
```

In [32]:
```
#set is immutable
```

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

Out[33]:
```
(78, -90.78, False, 'String', 'T')
```

In [34]:
```
s1+s2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13880\1145531307.py in <module>
----> 1 s1+s2

TypeError: can only concatenate tuple (not "set") to tuple
```

In [35]:
```
list3
```

Out[35]:
```
[1, 1, 1, 2, 1, 3, 2, 1, 3, 4]
```

In [36]:
```
list3=set(list3)
list3
```

Out[36]:
```
{1, 2, 3, 4}
```

In [ ]:
```
### SET PROPERTIES
1.Unorederd
2.Unindexed
3.Addition of item is possible but not in the given index directly
4.Duplicate items are not allowed (Unique elements only)
```

Tuple

In [37]:
```
# Empty tuple
```

In [38]:
```
mytup=()
mytup
```

Out[38]:
```
()
```

In [39]:
```
type(mytup)
```

Out[39]:
```
tuple
```

In [40]:
```
print(dir(mytup),end=" ")
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']
```

In [42]:
```
tuple1=(1, 1, 1, 2, 1, 3, 2, 1, 3, 4)
tuple1.count(1)
```

Out[42]:
```
5
```

In [43]:
```
tuple1
```

Out[43]:
```
(1, 1, 1, 2, 1, 3, 2, 1, 3, 4)
```

In [44]:
```
tuple1.index(1)
```

Out[44]:
```
0
```

In [45]:
```
tuple1.index(2)
```

Out[45]:
```
3
```

In [46]:
```
t1=("we",34,90,True)
t2=("Kindly","be","attentive","last",12,"stdents")
t1+t2
```

Out[46]:
```
('we', 34, 90, True, 'Kindly', 'be', 'attentive', 'last', 12, 'stdents')
```

In [47]:
```
t1[3]
```

Out[47]:
```
True
```

In [48]:
```
t2[3]
```

Out[48]:
```
'last'
```

In [49]:
```
t2[3]=78
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13880\1852230264.py in <module>
----> 1 t2[3]=78

TypeError: 'tuple' object does not support item assignment
```

In [59]:
```
print(t2[3:6])
```

```
('last', 12, 'stdents')
```

In [55]:
```
t2
```

Out[55]:
```
('Kindly', 'be', 'attentive', 'last', 12, 'stdents')
```

In [56]:
```
len(t2)
```

Out[56]:
```
6
```

In [57]:
```
print(t2[2:5])
```

```
('attentive', 'last', 12)
```

In [ ]:
```
#### PROPERTIES OF TUPLE
'''
1.Duplicates allowed
2.Ordered
3.Indexing allowed to access elements only
4.Immutable / Unmodifiable
5.Concatenation of tuples allowed(but not in set)
'''
```

Dictonary

In [60]:
```
d1={}
d1
```

Out[60]:
```
{}
```

In [61]:
```
d2=dict()
d2
```

Out[61]:
```
{}
```

In [62]:
```
type(d1),type(d2)
```

Out[62]:
```
(dict, dict)
```

In [63]:
```
stud={
    "sec":"IT-1",
    "sub":"TTL",
    "no_of_studs":78
}
```

In [64]:
```
print(stud) # key:value Pair
```

```
{'sec': 'IT-1', 'sub': 'TTL', 'no_of_studs': 78}
```

In [65]:
```
stud.keys()
```

Out[65]:
```
dict_keys(['sec', 'sub', 'no_of_studs'])
```

In [67]:
```
stud.values()
```

Out[67]:
```
dict_values(['IT-1', 'TTL', 78])
```

In [73]:
```
print(dir(dict),end=" ")
```

```
['__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__',
'__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__get
item__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__ior__', '__iter
__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__reduce__',
'__reduce_ex__', '__repr__', '__reversed__', '__ror__', '__setattr__', '__setitem_
_', '__sizeof__', '__str__', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'ge
t', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [74]:
```
stud.items()
```

Out[74]:
```
dict_items([('sec', 'IT-1'), ('sub', 'TTL'), ('no_of_studs', 78)])
```

In [75]:
```
for i in stud.items():
    print(i)
```

```
('sec', 'IT-1')
('sub', 'TTL')
('no_of_studs', 78)
```

In [76]:
```
stud["college"]="kiit"
stud
```

Out[76]:
```
{'sec': 'IT-1', 'sub': 'TTL', 'no_of_studs': 78, 'college': 'kiit'}
```

In [ ]:
```
### Properties of Dictionary
1.Mutable
2.Oredered
3.Duplicate keys not allowed
4.Values can be duplicate
5.No Indexing
```

In [77]:
```
### Dict constructor
mydict=dict(match = 'cricket',sport='hockey',year=2023)
mydict
```

Out[77]:
```
{'match': 'cricket', 'sport': 'hockey', 'year': 2023}
```

In [80]:
```
mydict['sport']
```

Out[80]:
```
'hockey'
```

In [81]:
```
del stud['sub']
stud
```

Out[81]:
```
{'sec': 'IT-1', 'no_of_studs': 78, 'college': 'kiit'}
```

In [ ]:
```

```