Today concepts

- sets
- Dictionaries

Sets

{}

- Advantage with sets is remove duplication
- Mostly sets gives the values in sorting

In [81]:
```
s = {1,2,3,4,5,1,2,3,}
type(s)
s
a
```

Out[81]:
```
10
```

In [6]:
```
l =  [1,2,4,2,3,"a","b","a","b"]  # [1, 2, 4, 3, 'a', 'b']
list(set(l))
```

Out[6]:
```
[1, 2, 3, 4, 'b', 'a']
```

In [2]:
```
s[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-2-c9c96910e542> in <module>
----> 1 s[0]

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

In [3]:
```
print(dir(set))
```

```
['__and__', '__class__', '__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__', '__x
or__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'i
ntersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'p
op', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union',
'update']
```

In [15]:
```
s1 = {"a",1,2,3,"b"}
s1.add("APSSDC")
s1
```

Out[15]:
```
{1, 2, 3, 'APSSDC', 'a', 'b'}
```

In [10]:
```
s1.add(s)

```

...

In [11]:
```
s = {}
type(s)
```

Out[11]:
```
dict
```

In [13]:
```
s = {}
t = set(s)
type(t)
```

Out[13]:
```
set
```

In [24]:
```
s1 = {1,2,3,4,"a","b"}
s2 = {3,4,"a","b",10,20}
```

In [25]:
```
print(s2.difference(s1))
print(s2)
print(s1)
```

```
{10, 20}
{3, 4, 10, 'b', 20, 'a'}
{1, 2, 3, 4, 'b', 'a'}
```

In [29]:
```
s2.difference_update(s1)
print(s2)
print(s1)
```

```
{10, 20}
{1, 2, 3, 4, 'b', 'a'}
```

In [30]:
```
s1 = {1,2,3,4,"a","b"}
s2 = {3,4,"a","b",10,20}
s1.intersection(s2)
```

Out[30]:
```
{3, 4, 'a', 'b'}
```

In [31]:
```
print(s2)
print(s1)
```

```
{3, 4, 10, 'b', 20, 'a'}
{1, 2, 3, 4, 'b', 'a'}
```

In [32]:
```
s1.intersection_update(s2)
print(s2)
print(s1)
```

```
{3, 4, 10, 'b', 20, 'a'}
{'b', 3, 4, 'a'}
```

In [35]:
```
s1 = {1,2,3,4,"a","b"}
s2 = {3,4,"a","b",10,20}
s1.isdisjoint(s2)
```

Out[35]: False

In [39]:
```
s1 = {1,2,3,4,5,6,7}
s2 = {3,2,1}
print(s1.issuperset(s2))
s2.issubset(s1)
```

```
True
```

Out[39]: True

In [44]:
```
# pop

s1.pop()
```

Out[44]: 3

In [53]:
```
s1
```

Out[53]: {4, 7}

In [57]:
```
s1 = {1,2,3,4,5,6,7}
print(s1.remove(7))
print(s1.discard(10))
print(s1.remove(10))

```

```
None
None

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-57-34e7d5adeea5> in <module>
      2 print(s1.remove(7))
      3 print(s1.discard(10))
----> 4 print(s1.remove(10))

KeyError: 10
```

In [58]:
```
s1
```

Out[58]:
```
{1, 2, 3, 4, 5, 6}
```

In [78]:
```
s1 = {1,2,3,4,"a","b"}
s2 = {3,4,"a","b",10,20}
print(s2.symmetric_difference(s1))
s1.symmetric_difference(s2)
```

```
{1, 2, 20, 10}
```

Out[78]:
```
{1, 2, 10, 20}
```

In [72]:
```
s2.symmetric_difference_update(s1)
s2
```

Out[72]:
```
{1, 2, 10, 20}
```

In [73]:
```
print(s1)
print(s2)
s1.union(s2)
```

```
{1, 2, 3, 4, 'b', 'a'}
{1, 2, 10, 20}
```

Out[73]:
```
{1, 10, 2, 20, 3, 4, 'a', 'b'}
```

In [74]:
```
print(s1)
print(s2)
```

```
{1, 2, 3, 4, 'b', 'a'}
{1, 2, 10, 20}
```

In [75]:
```
1 s1.update(s2)
```

In [76]:
```
1 s1
```

Out[76]:
```
{1, 10, 2, 20, 3, 4, 'a', 'b'}
```

In [77]:
```
1 print(s1)
2 print(s2)
3
```

```
{1, 2, 3, 4, 10, 'b', 20, 'a'}
{1, 2, 10, 20}
```

In [83]:
```
1 a = 10
2 a
```

Out[83]:
```
10
```

In [80]:
```
1 b = 20
2 b
```

Out[80]:
```
20
```

In [82]:
```
1 a = 30
2 a
```

Out[82]:
```
30
```

In [84]:
```
1 a
```

Out[84]:
```
10
```

Dictionaries

{key:value}

- we can assign any data type or any data structure in value position

In [85]:
```
1 d ={"a":"Apssdc","b":"Python",1:"Programming",2:["a","b","c"]}
2 d
```

Out[85]:
```
{'a': 'Apssdc', 'b': 'Python', 1: 'Programming', 2: ['a', 'b', 'c']}
```

In [86]:
```
1 type(d)
```

Out[86]:
```
dict
```

In [87]:
```
d ={"a":"Apssdc","b":"Python",1:"Programming",2:["a","b","c"],"b":20}
d
```

Out[87]:
```
{'a': 'Apssdc', 'b': 20, 1: 'Programming', 2: ['a', 'b', 'c']}
```

In [88]:
```
d[1]
```

Out[88]:
```
'Programming'
```

In [93]:
```
d[2]

```

Out[93]:
```
['a', 'b', 'c']
```

In [94]:
```
d[2][2]
```

Out[94]:
```
'c'
```

In [95]:
```
d = {1:10,2:10,3:10}
```

In [96]:
```
d
```

Out[96]:
```
{1: 10, 2: 10, 3: 10}
```

In [97]:
```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt_
_', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len_
_', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'cle
ar', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefaul
t', 'update', 'values']
```

In [98]:
```
d
```

Out[98]:
```
{1: 10, 2: 10, 3: 10}
```

In [99]:
```
d.keys()
```

Out[99]:
```
dict_keys([1, 2, 3])
```

In [100]:
```
d.values()
```

Out[100]:
```
dict_values([10, 10, 10])
```

In [101]:
```
d.items()
```

Out[101]:
```
dict_items([(1, 10), (2, 10), (3, 10)])
```

In [104]:
```
d ={"a":"Apssdc","b":"Python",1:"Programming",2:["a","b","c"],"b":20}
d.pop(1)
```

Out[104]: 'Programming'

In [105]:
```
d
```

Out[105]: {'a': 'Apssdc', 'b': 20, 2: ['a', 'b', 'c']}

In [106]:
```
d["MITS"] = "college"
d
```

Out[106]: {'a': 'Apssdc', 'b': 20, 2: ['a', 'b', 'c'], 'MITS': 'college'}

In [107]:
```
d[2] ="List"
d
```

Out[107]: {'a': 'Apssdc', 'b': 20, 2: 'List', 'MITS': 'college'}

In [108]:
```
d
```

Out[108]: {'a': 'Apssdc', 'b': 20, 2: 'List', 'MITS': 'college'}

In [112]:
```
d.popitem()

```

Out[112]: (2, 'List')

In [113]:
```
d
```

Out[113]: {'a': 'Apssdc', 'b': 20}

In [115]:
```
d.get("a")
```

Out[115]: 'Apssdc'

In [116]:
```
d["a"]
```

Out[116]: 'Apssdc'

In [117]:
```
d.setdefault("d")
d
```

Out[117]: {'a': 'Apssdc', 'b': 20, 'd': None}

In [118]:
```
d["d"]= "value"
```

In [120]:
```
d.setdefault("default","d")
d
```

Out[120]:
```
{'a': 'Apssdc', 'b': 20, 'd': 'value', 'default': 'd'}
```

In [133]:
```
l = [1,2,3,4]
set(l)
```

Out[133]:
```
{1, 2, 3, 4}
```

In [139]:
```
l1 = ["a","b"]
d = dict.fromkeys(l,l1)
d

```

Out[139]:
```
{1: ['a', 'b'], 2: ['a', 'b'], 3: ['a', 'b'], 4: ['a', 'b']}
```

In [140]:
```
d[5] = "MITS"
d
```

Out[140]:
```
{1: ['a', 'b'], 2: ['a', 'b'], 3: ['a', 'b'], 4: ['a', 'b'], 5: 'MITS'}
```

In [144]:
```
d.update({2:"Apssdc",10:30})
```

In [ ]:
```

```

In [145]:
```
d
```

Out[145]:
```
{1: ['a', 'b'], 2: 'Apssdc', 3: ['a', 'b'], 4: ['a', 'b'], 5: 'MITS', 10: 30}
```

In [ ]:
```

```