PYTHON FROZENSET ()

The frozenset () function returns an immutable frozenset object initialized with elements from the given iterable. Frozen set is just an immutable version of a Python set object. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation. Due to this, frozen sets can be used as keys in Dictionary or as elements of another set. But like sets, it is not ordered (the elements can be set at any index).

In Python, frozenset is the same as set except the frozen sets are immutable which means that elements from the frozenset cannot be added or removed once created

The syntax of frozenset () function is:

frozenset([iterable])

FROZENSET () PARAMETERS

The frozenset () function takes a single parameter:

iterable (Optional) - the iterable which contains elements to initialize the frozenset with. Iterable can be set, dictionary, tuple, etc.

RETURN VALUE FROM FROZENSET ()

The frozenset () function returns an immutable frozenset initialized with elements from the given iterable. If no parameters are passed, it returns an empty frozenset.

WORKING OF PYTHON FROZENSET ()

Frozenset on a tuple

names=("Zato","Gilbert","Maxwell","Samuel","Christian")

>>> fronzen\_set=frozenset(names)

>>> print ("The frozen set is: ",fronzen\_set)

The frozen set is: frozenset ({'Christian', 'Maxwell', 'Gilbert', 'Zato', 'Samuel'})

Frozenset on a dictionary

# random dictionary

person = {"name": "John", "age": 23, "sex": "male"}

fSet = frozenset(person)

print ('The frozen set is:', fSet)

The frozen set is: frozenset ({'name', 'sex', 'age'})

DIFFERNCE BETWEEN ORDERED DICTIONARY AND A NORMAL DICTIONARY

An OrderedDict is a dictionary subclass that remembers the order that keys were first inserted. The only difference between dict() and OrderedDict() is that:

OrderedDict preserves the order in which the keys are inserted. A regular dict doesn’t track the insertion order, and iterating it gives the values in an arbitrary order. By contrast, the order the items are inserted is remembered by OrderedDict.

EXAMPLE.

import collections

# we will first create normal dict

print('Dict:')

user\_dict = {}

user\_dict['PP'] = 10

user\_dict['QQ'] = 20

user\_dict['RR'] = 30

user\_dict['SS'] = 40

user\_dict['TT'] = 50

user\_dict['UU'] = 60

for item in user\_dict.items():

print(item)

print()

# now, we will create ordered dict

print('OrderedDict:')

user\_ordered\_dict = collections.OrderedDict()

user\_ordered\_dict['PP'] = 10

user\_ordered\_dict['QQ'] = 20

user\_ordered\_dict['RR'] = 30

user\_ordered\_dict['SS'] = 40

user\_ordered\_dict['TT'] = 50

user\_ordered\_dict['UU'] = 60

for item in user\_ordered\_dict.items():

print(item)

output

Dict:

('PP', 10)

('QQ', 20)

('RR', 30)

('SS', 40)

('TT', 50)

('UU', 60)

OrderedDict:

('PP', 10)

('QQ', 20)

('RR', 30)

('SS', 40)

('TT', 50)

('UU', 60)