**OrderedDict in Python**

An **OrderedDict** is a dictionary subclass that remembers the order that keys were first inserted. The only difference between [dict()](http://quiz.geeksforgeeks.org/python-set-4-dictionary-keywords-python/) 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.

# A Python program to demonstrate working of OrderedDict

from collections import OrderedDict

print("This is a Dict:\n")

d = {}

d['a'] = 1

d['b'] = 2

d['c'] = 3

d['d'] = 4

for key, value in d.items():

    print(key, value)

print("\nThis is an Ordered Dict:\n")

od = OrderedDict()

od['a'] = 1

od['b'] = 2

od['c'] = 3

od['d'] = 4

for key, value in od.items():

    print(key, value)

**Key value Change:** If the value of a certain key is changed, the position of the key remains unchanged in OrderedDict.

# A Python program to demonstrate working of key

# value change in OrderedDict

from collections import OrderedDict

print("Before:\n")

od = OrderedDict()

od['a'] = 1

od['b'] = 2

od['c'] = 3

od['d'] = 4

for key, value in od.items():

    print(key, value)

print("\nAfter:\n")

od['c'] = 5

for key, value in od.items():

    print(key, value)

**Deletion and Re-Inserting**: Deleting and re-inserting the same key will push it to the back as OrderedDict however maintains the order of insertion.

# A Python program to demonstrate working of deletion

# re-inserion in OrderedDict

from collections import OrderedDict

print("Before deleting:\n")

od = OrderedDict()

od['a'] = 1

od['b'] = 2

od['c'] = 3

od['d'] = 4

for key, value in od.items():

    print(key, value)

print("\nAfter deleting:\n")

od.pop('c')

for key, value in od.items():

    print(key, value)

print("\nAfter re-inserting:\n")

od['c'] = 3

for key, value in od.items():

    print(key, value)