**ChainMap in Python**

Python contains a container called “**ChainMap**” which encapsulates many [dictionaries](http://quiz.geeksforgeeks.org/python-set-4-dictionary-keywords-python/) into one unit. ChainMap is member of module “[**collections**](https://www.geeksforgeeks.org/python-collections-module/)“.

**Example:**

|  |
| --- |
| # Python program to demonstrate  # ChainMap      from collections import ChainMap      d1 = {'a': 1, 'b': 2}  d2 = {'c': 3, 'd': 4}  d3 = {'e': 5, 'f': 6}    # Defining the chainmap  c = ChainMap(d1, d2, d3)    print(c) |

 **keys() :-** This function is used to display all the **keys** of all the dictionaries in ChainMap.

 **values() :-** This function is used to display **values** of all the dictionaries in ChainMap.

 **maps() :-** This function is used to display **keys with corresponding values** of all the dictionaries in ChainMap.

# Python code to demonstrate ChainMap and

# importing collections for ChainMap operations

import collections

# initializing dictionaries

dic1 = { 'a' : 1, 'b' : 2 }

dic2 = { 'b' : 3, 'c' : 4 }

# initializing ChainMap

chain = collections.ChainMap(dic1, dic2)

# printing chainMap using maps

print ("All the ChainMap contents are : ")

print (chain.maps)

# printing keys using keys()

print ("All keys of ChainMap are : ")

print (list(chain.keys()))

# printing keys using keys()

print ("All values of ChainMap are : ")

print (list(chain.values()))

## Manipulating Operations

* **new\_child() :-** This function adds a new dictionary in the beginning of the ChainMap.
* **reversed() :-** This function reverses the relative ordering of dictionaries in the ChainMap.

# Python code to demonstrate ChainMap and

# reversed() and new\_child()

# importing collections for ChainMap operations

import collections

# initializing dictionaries

dic1 = { 'a' : 1, 'b' : 2 }

dic2 = { 'b' : 3, 'c' : 4 }

dic3 = { 'f' : 5 }

# initializing ChainMap

chain = collections.ChainMap(dic1, dic2)

# printing chainMap using map

print ("All the ChainMap contents are : ")

print (chain.maps)

# using new\_child() to add new dictionary

chain1 = chain.new\_child(dic3)

# printing chainMap using map

print ("Displaying new ChainMap : ")

print (chain1.maps)

# displaying value associated with b before reversing

print ("Value associated with b before reversing is : ",end="")

print (chain1['b'])

# reversing the ChainMap

chain1.maps = reversed(chain1.maps)

# displaying value associated with b after reversing

print ("Value associated with b after reversing is : ",end="")

print (chain1['b'])