**Counters in Python**

**Counter** is a **container** included in the [collections](https://www.geeksforgeeks.org/python-collections-module/) module. Now you all must be wondering what is a container.

A Counter is a subclass of dict. Therefore it is an unordered collection where elements and their respective count are stored as a dictionary. This is equivalent to a bag or multiset of other languages.

**Syntax :**

class collections.Counter([iterable-or-mapping])

The constructor of counter can be called in any one of the following ways :

 With sequence of items

 With dictionary containing keys and counts

 With keyword arguments mapping string names to counts

**Example of each type of initialization :**

|  |
| --- |
| # A Python program to show different ways to create  # Counter  from collections import Counter    # With sequence of items  print(Counter(['B','B','A','B','C','A','B','B','A','C']))    # with dictionary  print(Counter({'A':6, 'B':7, 'C':8}))    # with keyword arguments  print(Counter(A=6, B=7, C=8))  **Updation :**  We can also create an empty counter in the following manner :  count = collections.Counter()  And can be updated via update() method .Syntax for the same :  coun.update(Data)  # A Python program to demonstrate update()  from collections import Counter  coun = Counter()    coun.update([1, 2, 3, 1, 2, 1, 1, 2])  print(coun)    coun.update([1, 2, 4])  print(coun)   Data can be provided in any of the three ways as mentioned in initialization and the counter’s data will be increased not replaced.   Counts can be zero and negative also.  # Python program to demonstrate that counts in  # Counter can be 0 and negative  from collections import Counter    c1 = Counter(A=4,  B=3, C=10)  c2 = Counter(A=10, B=3, C=4)    c1.subtract(c2)  print(c1)   We can use Counter to count distinct elements of a list or other collections.  # An example program where different list items are  # counted using counter  from collections import Counter    # Create a list  z = ['blue', 'red', 'blue', 'yellow', 'blue', 'red']    # Count distinct elements and print Counter aboject  print(Counter(z) |