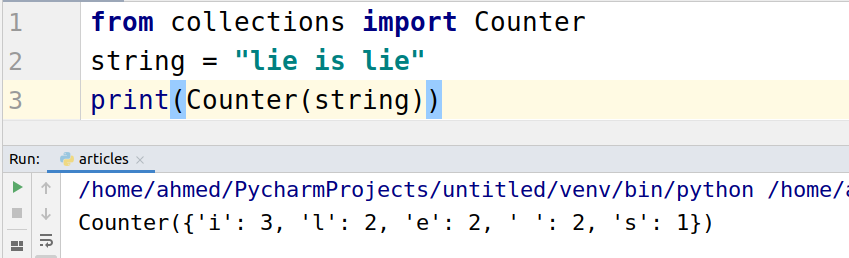
Python Counter in Collections with Example

Python counter helps us count the number of elements repeated in the list, dictionary, tuple or any data structure. For example, if you have a string “hello world”, the counter will return a dict whose key will be the alphabet and the value will be the number of times that alphabet repeated. Let’s take a look at code.

## Counter with String

|  |
| --- |
| from collections import Counter string = "lie is lie" print(Counter(string)) |

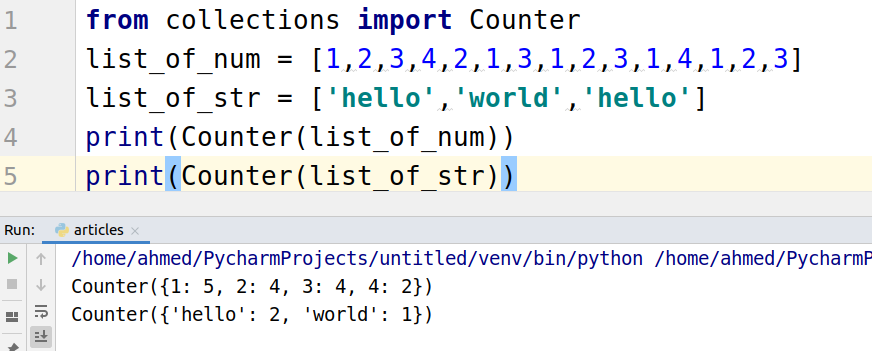
The following is the output.



## Counter with List

The counter function treats the list the same as string. It takes the list values as input and outputs the dictionary with the key as the elements of the list and value as the count of the repeated element.

|  |
| --- |
| from collections import Counter list\_of\_num = [1,2,3,4,2,1,3,1,2,3,1,4,1,2,3] list\_of\_str = ['hello','world','hello'] print(Counter(list\_of\_num)) print(Counter(list\_of\_str)) |

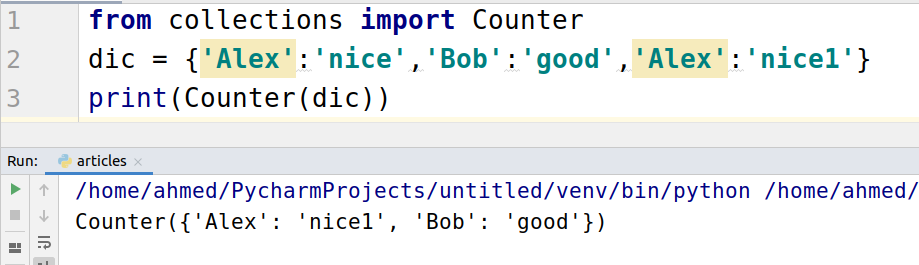


## Counter with Dictionary

When you pass a dictionary to the counter function, it will remove the redundant elements in the dictionary.

Let’s take a look at code.

|  |
| --- |
| from collections import Counter dic = {'Alex':'nice','Bob':'good','Alex':'nice1'} print(Counter(dic)) |

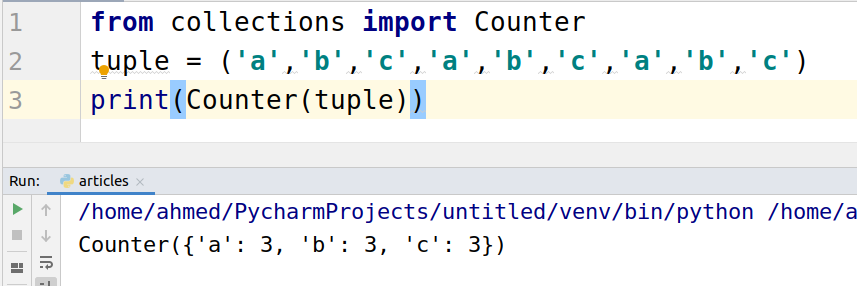


## Counter with Tuple

The counter function works the same for the tuple as it works for the string. It counts each element present in tuple then put it inside the dict as key-value pairs.

Let’s take a look at code.

|  |
| --- |
| from collections import Counter tuple = ('a','b','c','a','b','c','a','b','c') print(Counter(tuple)) |



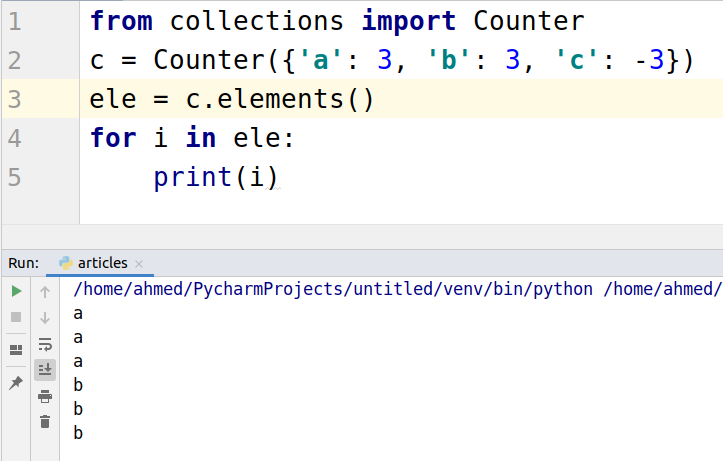
## Methods Available on Python Counter

### Counter.elemets()

This function returns all the values with a positive value.

Let’s take a look at code.

|  |
| --- |
| from collections import Counter c = Counter({'a': 3, 'b': 3, 'c': -3}) ele = c.elements() for i in ele:  print(i) |



### counter.most\_common(value)

The most\_common function returns the most common value present in counter. It takes a value as input. This value determines how many top values you want.

|  |
| --- |
| from collections import Counter c = Counter({'a': 1, 'b': 2, 'c': 3}) ele = c.most\_common(1) print(ele) ele = c.most\_common(2) print(ele) |

