

Custom Search

Courses

Write an Article



Python Tricks for Competitive Coding

Python is one such programming language which makes everything easier and straight forward. Anyone who has dabbled in python for Competitive Coding gets somewhat addicted to its many features. Here is list of some of its cool features that that I've found most useful in a competitive coding environment.

1. The `most_common` function of the `Counter` Package.

This is probably the most useful function I've ever used and its always at the back of my mind while writing any python code. This function analyses a list/string and helps to return the top `n` entities in the list/string according to their number of occurrences in descending order where `n` is a number that is specified by the programmer. The individual entities are returned along with their number of occurrences in a **tuple** which can easily be referred/printed as and when required.

```
# Code to find top 3 elements and their counts
# using most_common
from collections import Counter

arr = [1, 3, 4, 1, 2, 1, 1, 3, 4, 3, 5, 1, 2, 5, 3, 4, 5]
counter = Counter(arr)
top_three = counter.most_common(3)
print(top_three)
```

Output:

```
[(1, 5), (3, 4), (4, 3)]
```

The output tuple clearly states that 1 has occurred 5 times, 3 has occurred 4 times, and 4 has occurred 3 times.

2. The n-largest/n-smallest function of the heapq Package.

This function helps to return the top **n** smallest/largest elements in any lists and here again **n** is a number specified by the programmer.

```
# Python code to find 3 largest and 4 smallest
# elements of a list.
import heapq

grades = [110, 25, 38, 49, 20, 95, 33, 87, 80, 90]
print(heapq.nlargest(3, grades))
print(heapq.nsmallest(4, grades))
```

Output:

```
[110, 95, 90]
[20, 25, 33, 38]
```

The first line of output gives 3 of the largest numbers present in the list grades. Similarly the second line of output prints out 4 of the smallest elements present in the list grades. Another **speciality** of this function is that it does not overlook repetitions. So in place of **n** if we were to place the length of the array the we would end up with the entire sorted array itself !!

3. Dictionary and concept of zipping Dictionaries

Dictionaries in python are truly fascinating in terms of the unique functionality that they offer. They are stored as a **Key and Value pair** in the form of an array like structure. Each value can be accessed by its corresponding key.

The zip function is used to join two lists together or we can even join the key and value pairs in a dictionary together as a single list. The application of this concept will be made clear in the following code snippet.

```
# Python code to demonstrate use of zip.
import heapq

stocks = {
    'Goog' : 520.54,
    'FB' : 76.45,
    'yhoo' : 39.28,
    'AMZN' : 306.21,
    'APPL' : 99.76
}


zipped_1 = zip(stocks.values(), stocks.keys())
```

```
# sorting according to values
print(sorted(zipped_1))

zipped_2 = zip(stocks.keys(), stocks.values())
print(sorted(zipped_2))
#sorting according to keys
```

Output:

```
[(39.28, 'yhoo'), (76.45, 'FB'), (99.76, 'APPL'), (306.21, 'AMZN')
[('AMZN', 306.21), ('APPL', 99.76), ('FB', 76.45), ('Goog', 520.5
```



4. The Map function.

This function is a sneaky little shortcut that allows us to implement a simple function on a list of values in a very **Unconventional Manner**. The following example will give a simple application of this functionality. The function takes as parameters the function name and the name of the list the function needs to be applied upon.

```
# Python code to apply a function on a list
income = [10, 30, 75]

def double_money(dollars):
    return dollars * 2

new_income = list(map(double_money, income))
print(new_income)
```

Output:

```
[20, 60, 150]
```

Here, we just implemented a simple function which multiplies each list value by two and returns it as a new list.

Individually these functions might look innocent but will definitely come in handy in a TIME LIMITED CODING ENVIRONMENT in the sense that they offer large functionality in a VERY short amount of code. The functionalities discussed have very specific applications and act like a SHORTCUT or a CHEAT-SHEET in competitive coding. Having these useful tricks up your sleeve might just give someone the COMPETITIVE EDGE that they were looking for !!

This article is contributed by **Siddhant Bajaj**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

[Tips and Tricks for Competitive Programmers | Set 2 \(Language to be used for Competitive Programming\)](#)

[Why is python best suited for Competitive Coding?](#)

[Bit Tricks for Competitive Programming](#)

[C++ tricks for competitive programming \(for C++ 11\)](#)

[Some useful C++ tricks for beginners in Competitive Programming](#)

[Tips and Tricks for Competitive Programmers | Set 1 \(For Beginners\)](#)

[BFS using STL for competitive coding](#)

[Java tricks for competitive programming \(for Java 8\)](#)

[Selenium Python Tricks](#)

[10 Interesting Python Cool Tricks](#)

[10 Essential Python Tips And Tricks For Programmers](#)

[Python in Competitive Programming](#)

[Python Input Methods for Competitive Programming](#)

[Input/Output from external file in C/C++, Java and Python for Competitive Programming](#)

[Input/Output from external file in C/C++, Java and Python for Competitive Programming | Set 2](#)

Article Tags : [Competitive Programming](#) [Python](#)



7

☐ To-do ☐ Done

2.3

Based on 3 vote(s)

[Feedback/ Suggest Improvement](#)[Notes](#)[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Share this post!](#)

0 Comments

GeeksforGeeks

 **Sudhanshu Joshi** ▾ **Recommend** **Tweet** **Share****Sort by Newest** ▾

Be the first to comment.

[!\[\]\(e119fc79c8f448683d20ba4c873025a2_img.jpg\) Subscribe](#) [!\[\]\(80c84d616db6940097dc4a95f6a78636_img.jpg\) Add Disqus to your site](#) [Add DisqusAdd](#) [!\[\]\(eed2983d93af5f80b80b6aed158b2294_img.jpg\) Disqus' Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#)

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved