# **Name:** Karin Sinkevitch

**Date:** August 17, 2020

**Course:** IT FDN 110A

**Assignment:** 06

**GitHub URL:** <https://github.com/KARSINK22/IntroToProg-Python-Mod06>

# **Functions, Simple Classes, and the PyCharm Debugger**

# Introduction

This document provides a brief introduction to Python functions, simple classes, and using the debugger in PyCharm. Reference information can be used to learn more and view examples on these topics. An example script is then described to show how these methods can be applied.

# Functions, Simple Classes, PyCharm Debugging Tool

# Functions

A function is a block of code which only runs when it is called. You can pass data, known as arguments, into a function. Functions can return data as a result. The structure of a function can be found in the references provided. A parameter is what is declared in the function and an argument is what is passed through when calling the function. It is useful to note that when returning multiple items from a function the data it is returned as a tuple. Function parameters can be given default values. If an argument is omitted when the function is called, the default parameter value is used. A function’s default values can also used with conditional statements inside the function to simulate overloaded functions which deal with different types of data arguments. Figure 1 and Figure 2 provide the syntax and an example of a function in Python.

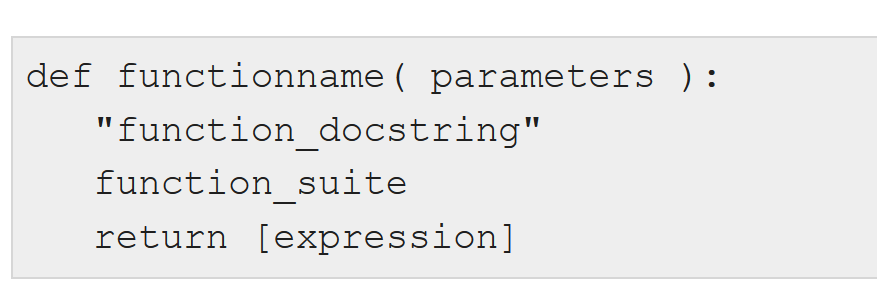


Figure 1 Python Function Syntax

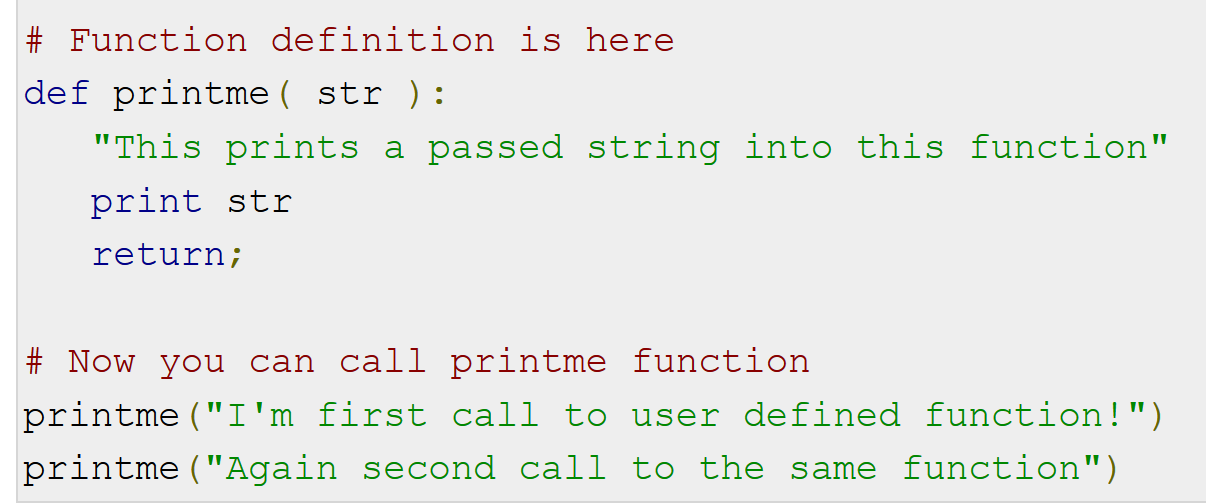


Figure 2 Python Function Example

Python uses a mechanism, which is known as "Call-by-Object", sometimes also called "Call by Object Reference" or "Call by Sharing". If you pass immutable arguments like integers, strings or tuples to a function, the passing acts like Call-by-value. It is different if mutable arguments are passed. All parameters (arguments) in the Python language are passed by reference. It means if you change what a parameter refers to within a function, the change also reflects back in the calling function.

(Referencehttps://www.w3schools.com/python/python\_functions.asp, <https://stackoverflow.com/questions/47169033/parameter-vs-argument-python>, and https://www.tutorialspoint.com/how-are-arguments-passed-by-value-or-by-reference-in-python).

# Global and Local Variables

Variables that are defined inside a function body have a local scope, and those defined outside have a global scope. This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions. When you call a function, the variables declared inside it are brought into scope.

Be careful of shadowing where a global variable is used inside a function and can be overwritten. Best practice is to not use global variable inside of functions but if they are used, declare the variable as global using the “global” keyword.

(Reference: <https://www.tutorialspoint.com/global-vs-local-variables-in-python> and Randal Root “\_Mod6PythonProgrammingNotes.pdf”).

# Using Functions to Organize Code

Functions help break our program into smaller and modular chunks. Functions also avoid repetition and make the code reusable and more shareable.

(References: https://www.programiz.com/python-programming/function)

# Simple Classes

Python is an object-oriented programming language. Almost everything in Python is an object with properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects. Classes can be used to group constants, variables, and functions. (Reference: https://www.w3schools.com/python/python\_classes.asp).

# Debugging in PyCharm

Debugging a script helps find errors and issues when simply running the code are hard to find. PyCharm makes debugging Python scripts easy. See the reference for a great walkthrough of using the PyCharm debugging tool. (Reference: https://www.youtube.com/watch?v=QJtWxm12Eo0).

# Create Python Script “Assigment06\_Starter.py”

This section describes how to create a script called “Assigment06\_Starter.py”. The script reads in an existing to do list called “ToDoFile.txt” which is a comma separated file with tasks and priorities (Figure 3). The list is read into a list of dictionaries. Each dictionary contains two values: task and priority. Try-except error handling is used when reading the file in case the file does not exist or there is an error when reading it.

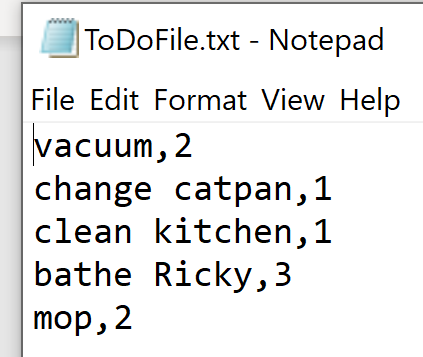


Figure 3 ToDoFile.txt

This script provides a great example of using functions to organize code into data, processing, and presentation. The functions are structured into classes which also help the organization and readability of the code. There are two classes: class Processor and class IO. The main body of the script is a while loop that uses the IO class functions to request and receive input from the user and the Processor class functions to manipulate the data based on that input.

Worth pointing out is the Processor class function “remove\_data\_from\_list”. This function removes a row from the list of dictionaries based on the value of the variable “task”. Because of the way Python handles lists, only the first row in the list that matches the variable will be deleted. This is a problem if the there are duplicate tasks in the list. Therefore, the function creates a new list that copies all dictionary rows from the original list of tasks except for the tasks that match the name of the task to remove (Figure 4).

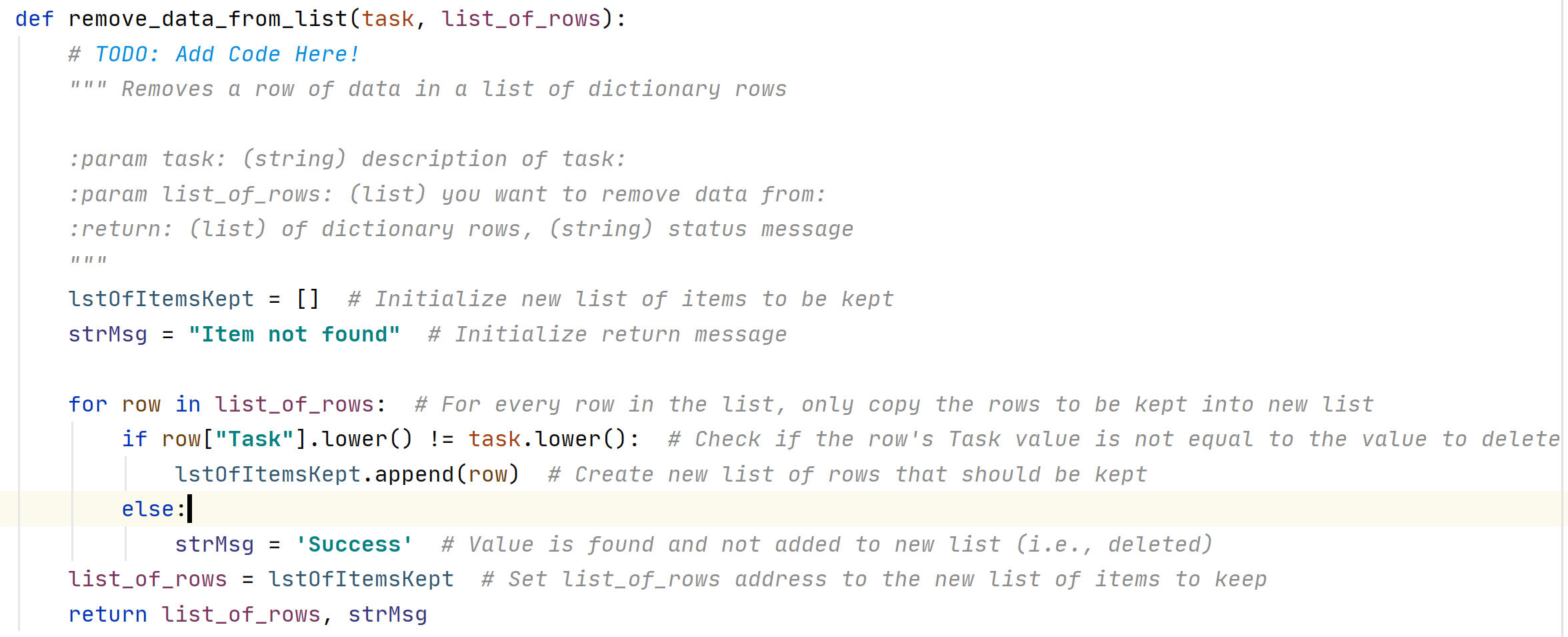


Figure 4 Function: remove\_data\_from\_list

Figure 5 through Figure 7 show the highlights of the code running in PyCharm.

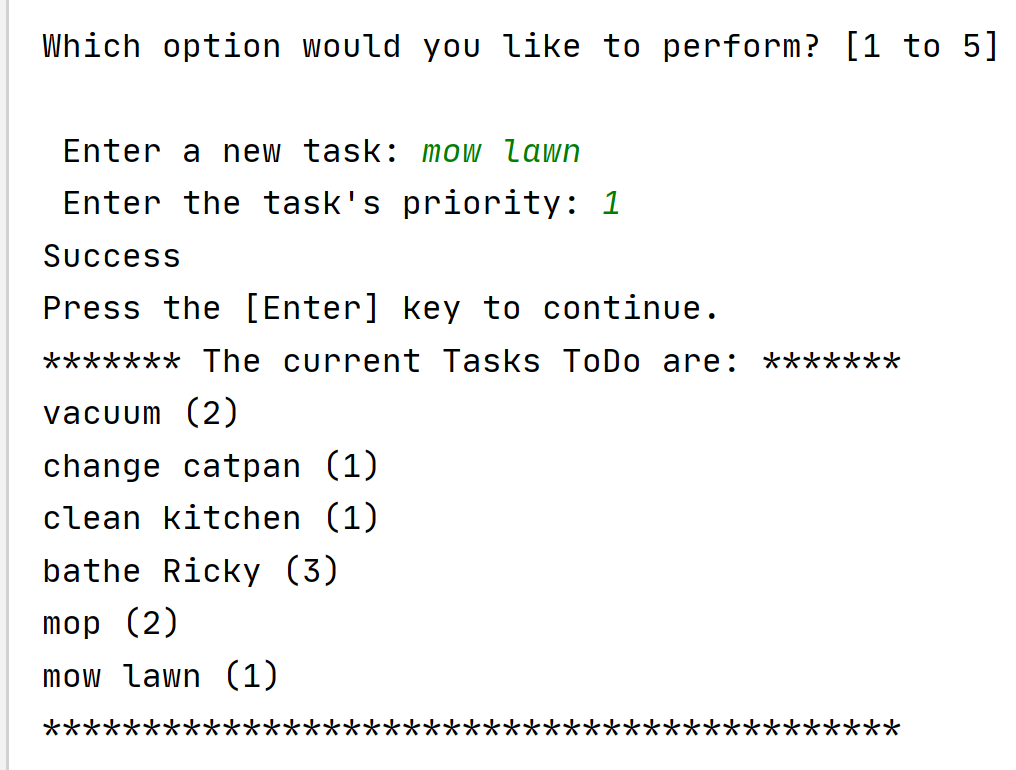


Figure 5 Adding a task to the list

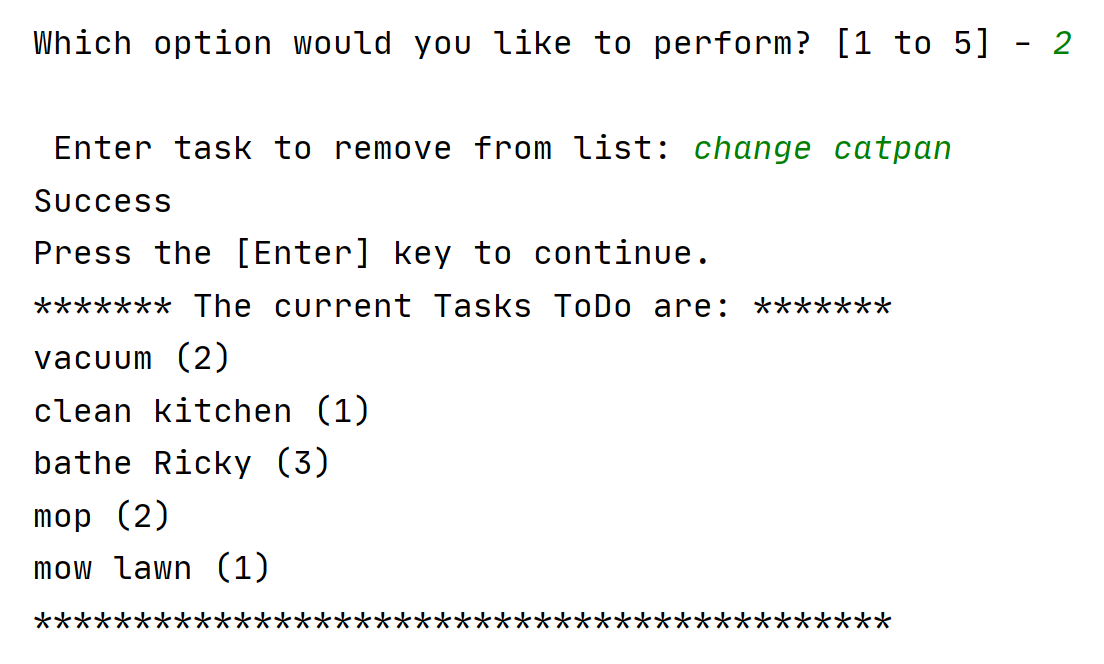
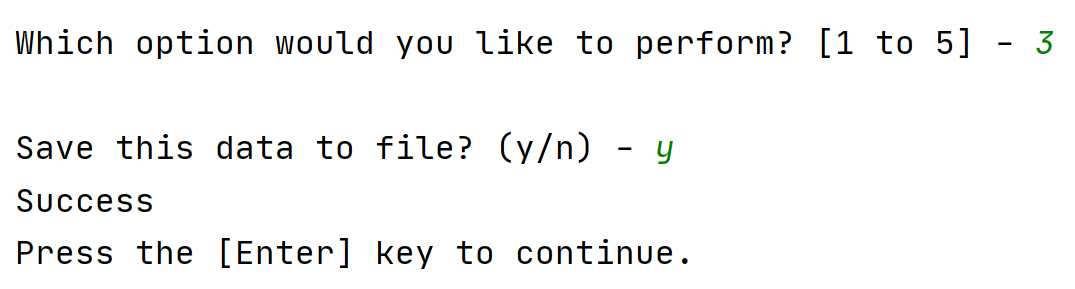


Figure 6 Removing a task from the list



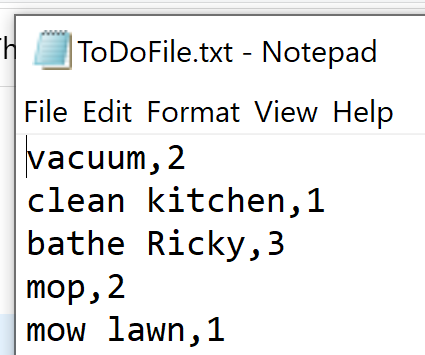


Figure 7 Saving list to a file and the result

Figure 8 shows the script being run in the console with the example of removing multiple tasks with the same name.

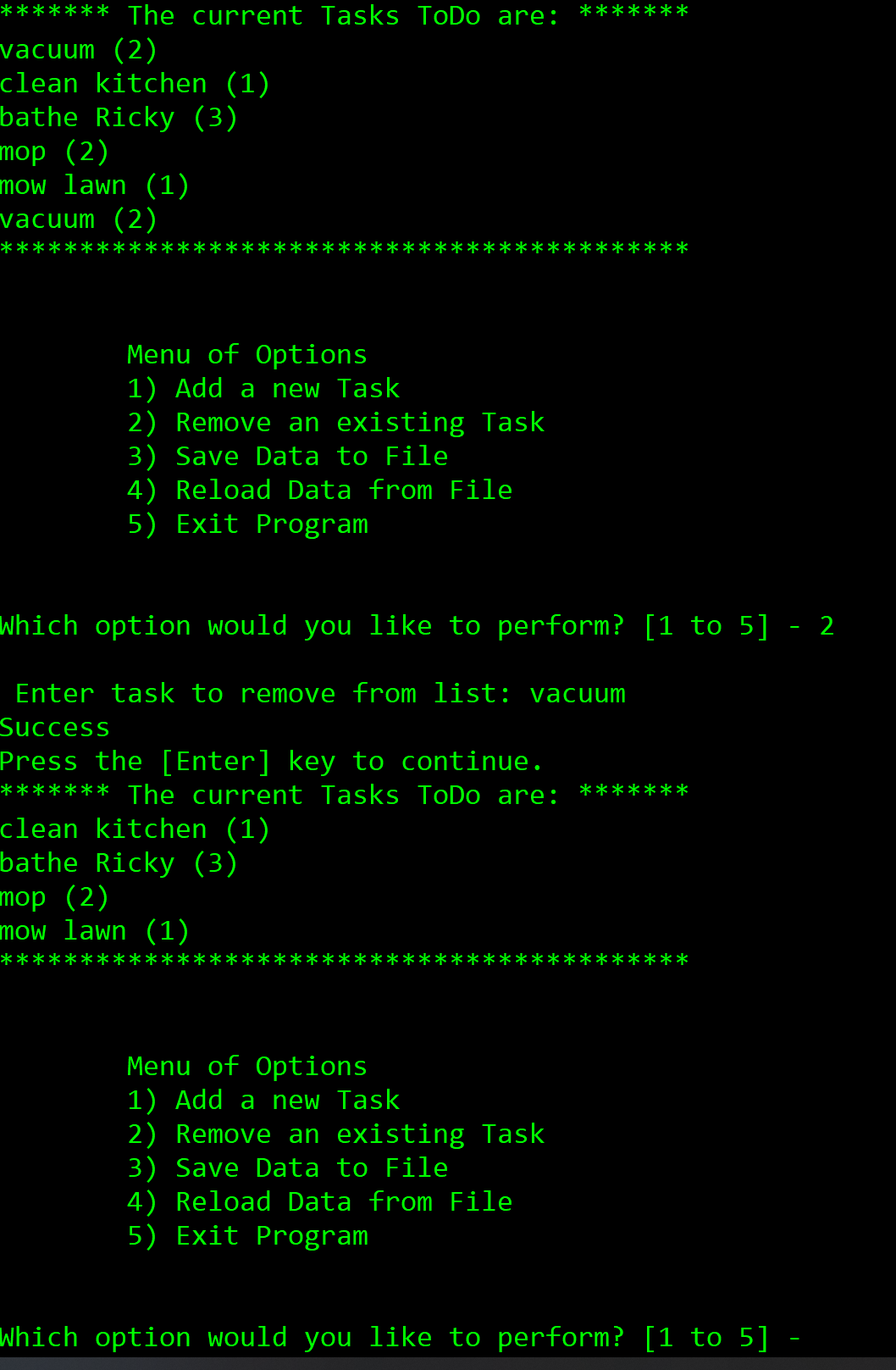


Figure 8 Script running in console

# Summary

Using functions and classes in scripts takes coding to the next level. When using functions in Python, it is important to understand how global and local variables work and how arguments are passed. PyCharm is a great development environment and the debugger is a great tool.