Daniel Medrano Figueora

05/22/2023

**IT FDN 110 A** 

Assignment 06

# **Operators, Expression, and Conditional Statements**

### Intro

This week we learned about global and local variables. Global variables in Python are defined outside of any function and can be accessed from anywhere in the program. They have a global scope and can be modified within functions. Local variables, on the other hand, are defined within a function and can only be accessed within that function. They have a local scope and their values are discarded once the function execution is complete. Classes in Python are blueprints for creating objects that encapsulate data and functionality, while functions are blocks of code that perform specific tasks and can be reused throughout the program. The debugger tool became useful as well to overlook the code and understand where critical issues are.

## **Background**

This week's assignment is to modify the Assignment06\_Starter file. Using the methods learned this week, the program is supposed to have a menu of options to add tasks, remove tasks/view them or to save the file. This is all done using classes to call and do the operations.

### Intro

I first modified the read data from the file. This is where the code is able to read from the file and is able to add the words task and priority so that it may save more data. The word task and priority gets added to the file to each one the user inputs and can be brought up at any time.

The class in the image below adds the data that uses inputs into the list.

```
destaticmethod
def add_data_to_list(task, priority, list_of_rows):
    """ Adds data to a list of dictionary rows

iparam task: (string) with name of task:
    iparam priority: (string) with name of priority:
    iparam list_of_rows: (list) you want to add more data to:
    ireturn: (list) of dictionary rows

"""

row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
list_of_rows.append(row)
return list_of_rows
```

The class below removes the tasks and priorities.

The class below writes data to the file

Each of these classes are called out throughout the code and help keep the code organized in knowing where specific operations are.

Moving onto the IO portion where the menu is operated.

The last two IO sections are displayed in the image below. The first input, gets the task and priority values and adds it to the list which returns a string and its respected task and priority. The second input task asks the user which task to remove from the list and does so by looking up the task.

```
destaticmethod
def input_new_task_and_priority():
    """ Gets task and priority values to be added to the list
    :return: (string, string) with task and priority
    """
    task = input("Enter a new task: ")
    priority = input("Enter the priority for the task: ")
    return task, priority

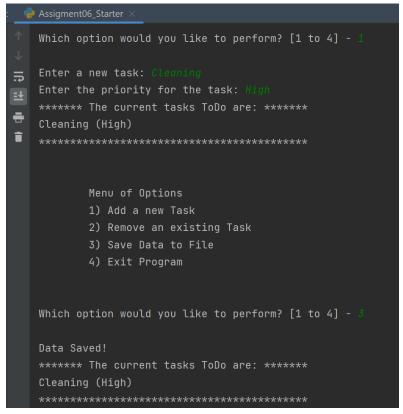
Gets the task name to be removed from the list

:return: (string) with task
    """

task = input("Enter the task to remove: ")
    return task
```

Lastly, the main body of the code is where most of the commands get processed. The previous code shown is more of temporary data storage and user inputs. The main body is now less code as the work gets broken up into the previous two sections which makes it a lot easier to understand for anyone working on the code.

Running the code within PyCharm, asking to create a new task of "cleaning" and the priority to be "high". Then Saving the code and exiting the program.



Using the terminal window to remove the current tasks.

The task of cleaning was removed successfully and was able to save and exit from the terminal window.

# **Summary**

In the end, global variables are defined outside of any function and can be accessed from anywhere in the program, while local variables are defined within a function and can only be accessed within that function. Classes are used to create objects that contain both data and functionality, providing a blueprint for their structure and behavior. Functions, on the other hand, are blocks of code that perform specific tasks and can be reused throughout the program. Understanding the difference between global and local variables, as well as the concept of classes and functions, is crucial for effective programming in Python.