Lesson 10 Final Project: Task Manager

Alberta "Albi" Kovatcheva

- This notebook includes the instructions for this project to explain the steps of creation.

In [1]:
```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug  4 18:10:53 2021

@author: albi
"""
tasks = [
    {'name' : 'Write email to Jan', 'completed' : True},
    {'name' : 'Sweep front porch', 'completed' : True},
    {'name' : 'Call mom', 'completed' : False}
]
```

Requirements

Below are the requirements for this project. There is some functionality already set up in your project. Currently, if you run the program in your VSCode terminal, you will see a menu of options, as shown below:

1. List the tasks
2. Add a task
3. Remove a task
4. Mark a task complete
5. Quit

What would you like to do? As the user, you are able to input the number 1 to see all tasks that currently live in the list of tasks (dictionaries) named tasks. The goal here is for you to write the functionality to add, remove, and mark a task complete by inputting 2, 3, or 4, respectively. Instructions are listed below.

Tip! Remember to pay attention to indenting! When defining a function, you should have a colon at the end of the line, and Python will automatically indent for you. If you're not seeing the next line indented, double check your code!

In [2]:
```
def list_tasks():
    for index, task in enumerate(tasks):
        print(str.format('{}: {} (Completed: {})', index, task['name'], task[
```

Add a Task

1. Within the already created add_task function, there is a variable named task_text which is the result of the input() function which prompts the user to type in something.

- This input() function is a new function that you haven't seen before. It is essentially the opposite of the print() function. This allows for the user to input text when called.

2. Create a dictionary that uses the task_text variable as the name and sets completed to False.

3. Add the dictionary to the tasks list.

4. Within the while loop, add an elif statement that checks if the variable decision (defined within the while loop that allows the user to input a number) is 2, call the add_task function.

In [3]:
```
def add_task():
    task_text = input('Please add a task: ')
    # Using the above task_text variable, create a dictionary named new_task
    new_task = {'name' : task_text, 'completed' : False}
    # Then, add new_task to the tasks list.
    tasks.append(new_task)
```

Remove a task

1. You will need a function to handle this.

2. When this function is run, list out the tasks.

- Hint! There is already a function that handles the listing of tasks.

3. Create a variable that uses input(). The user should be able to input the index number of the task to be removed.

- Hint! You will need to wrap the input() function within the int() function so the user's input is read as a number.

4. Write the functionality to be able to delete the task in the list tasks based on the variable you created above.

5. Within the while loop, add an elif statement that allows your new function to be run when the correct menu option is chosen.

In [4]:
```
# You will need a function to handle removing a task.
def remove_task():
    # When this function is run, list out the tasks. Hint! There is already a
    list_tasks()
    # Here, create a variable that uses input. The user should be able to inp
    # Hint! You will need to wrap the input() function within the int() funct
    remove_todo = int(input('Please enter the number of the task to be remove
    # Here, delete the task in the tasks list based on the above variable.
    tasks.remove(tasks[remove_todo])
```

Mark a task complete

1. You will need a function to handle this.

2. When this function is run, list out the tasks.

- Hint! There is already a function that handles the listing of tasks.

3. Create a variable that uses input. The user should be able to input the index number of the task to be marked complete.

- Hint! You will need to wrap the input() function within the int() function so the user's input is read as a number.

4. Mark the task as complete in the list tasks based on the variable you created above.

- Hint! you will need to use two sets of square brackets to find the index and set the appropriate key to True.

5. Within the while loop, add an elif statement that allows your new function to be run when the correct menu option is chosen.

In [5]:
```
# You will need a function to handle marking a task complete.
def mark_complete():
    # When this function is run, list out the tasks. Hint! There is a functio
    list_tasks()
    # Here, create a variable that uses input.
    # The user should be able to input the index number of the task to be mar
    # Hint! You will need to wrap the input() function within the int() funct
    completed = int(input('Please enter the number of the task to be marked c
    # Mark the task complete in the tasks list based on the variable you crea
    # Hint! you will need to use two sets of square brackets to find the inde
    tasks[completed]['completed'] = True
```

In [6]:
```
menu_text = """
====================
1. List the tasks
2. Add a task
3. Remove a task
4. Mark task complete
5. Quit

What would you like to do? """
```

In [*]:

```
program_is_running = True
# Add elif statements for inputs 2, 3, and 4
while program_is_running:
    decision = input(menu_text)
    if decision == '1':
        list_tasks()
    elif decision == '2':
        add_task()
    elif decision == '3':
        remove_task()
    elif decision == '4':
        mark_complete()
    elif decision == '5':
        program_is_running = False
    else:
        print('please choose a valid option')
```

```
====================
1. List the tasks
2. Add a task
3. Remove a task
4. Mark task complete
5. Quit

What would you like to do?
```