Bita Massoudi

11/20/2022

Foundations of Programming Python

Assignment 06

**Python Functions and Classes**

**Introduction:**   
This week, functions and classes were introduced. Functions are a way to group statements as well as to make the group available by a suitable name defined (by the programmer). Functions allow for the passage of values for processing through arguments. In Python, we use classes to create objects. A class is a tool, like a blueprint or a template, for creating objects. It allows us to bundle data and functionality together.

The body of the script is the ‘if’ statements under the “while true” condition that determine the function to call based on the user’s input, such as loading a file, adding to a file, etc. Each of these functionalities requested by the user requires a separate function that can be called to execute the functionality. As a result, I developed 7 different functions. Fig 1, illustrates the if statements under the “while True “ statement.

Text

Description automatically generated

Fig 1. If statements and while True statements

## Using functions to load from a list:

This function reads from the text file and loads the data into Python memory into the table. Fig 2. Illustrates the load file script and output.

Text

Description automatically generated

Fig 2 a. def read\_file(file\_name, table). Load file script.

Load script

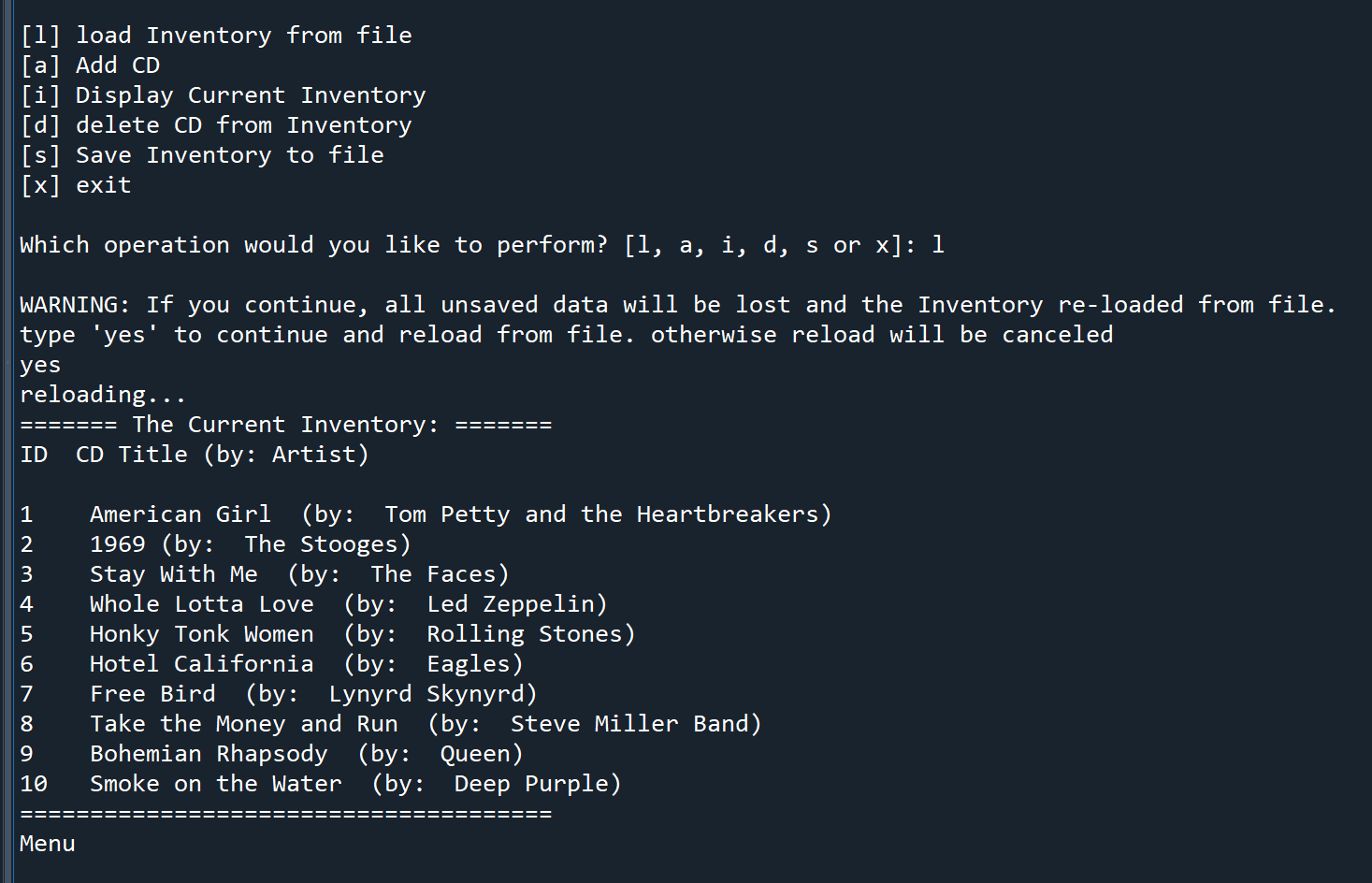


Fig 2 b. Output of load function

## Using functions to add items to inventory:

This function takes user input and appends it to the existing information loaded from the text file. The final outcome is a list of existing items from the text file plus items added by the user. Fig 3 a, b, c. Illustrates add script and output.

Text

Description automatically generated

Fig 3 a. Add script

Text

Description automatically generated

Fig 3 b. Append script

Text

Description automatically generated

Fig 3c. Add output

## Using functions to display a list:

This function displays the information loaded from the text file plus items added from the user. The output is a complete list of all items from the CD inventory. Fig 4 a, b illustrates display script and output.

Text

Description automatically generated

Fig 4a. Display script

Text

Description automatically generated

Fig 4 b. Display output

## Using functions to save to the .txt file:

This function shows the inventory to the user and asks the user to confirm to save the contents to the text file. If the user confirms, the contents are saves to text file. Fig 5 a, b and c. illustrates save function script and output.

Text

Description automatically generated

Fig 5a. Displays the inventory to users

Text

Description automatically generated

Fig 5b. Saves the items to a text file and updates the text file

Text

Description automatically generated

Fig 5c. Saved items are written to the text file and confirmed by the script

## Using functions to delete from a list:

This function initially identifies the row to be deleted. If the row is not found, the script skips deleting and messages the user. If it identifies the row requested by the user, then the row is deleted and confirms deletion to the user. Fig 6 a, b, c. illustrates delete function script and output.

**Function: Delete File [def delete\_file (intIDDel):]**

Text

Description automatically generated

Fig 6a. Delete number within range (CD is deleted)

Text

Description automatically generated

Fig 6 b. Delete number outside the range (could not find the CD)

Text

Description automatically generated

Fig 6 c. Delete output

## Exit:

Exit does not have a function. It is a break in the if statements. Fig 7 shows the exit output.

Text

Description automatically generated with medium confidence

## Conclusion

Functions create an experience that is easier to manage for the developer. Each function performs it’s own specialized tasks and it’s easier to track the work flow in a large software effort. On the other hand, it’s important to track the variables, lists and dictionaries between the functions to ensure that the flow of information between the functions remains consistent. It’s also easier to debug each function individually while one must keep the role of the function with respect to interaction with other functions and within the overall flow of the program clear.