

Programming Assignment

So far you have been writing programs to do specific tasks such as pickling, building linked lists etc and this task is to build an algorithm that manages lists of things (e.g. movie titles) stored in files.

It will involve using classes, python data structures and functions.

Write an interactive program that maintains lists of strings in files.

When the program is run it should create a list of all the files in the current directory that have the .lst extension.

Use `os.listdir(".")` to get all the files and filter out those that don't have the .lst extension.

If there are no matching files the program should prompt the user to enter a filename adding the .lst extension if the user doesn't enter it.

If there are one or more .lst files they should be printed as a numbered list starting from 1.

The user should be asked to enter the number of the file they want to load, or 0, in which case they should be asked to give a filename for a new file.

If an existing file was specified its items should be read.

If the file is empty, or if a new file was specified, the program should show a message, "no items are in the list".

If there are no items, two options should be offered: "Add" and "Quit".

Once the list has one or more items, the list should be shown with each item numbered from 1, and the options offered should be "Add", "Delete", "Save" (unless already saved), and "Quit".

If the user chooses "Quit" and there are unsaved changes they should be given the chance to save.

Here is a transcript of a session with the program (with most blank lines removed, and without the "List Keeper" title shown above the list each time):

```
Choose filename: movies
-- no items are in the list --
[A]dd [Q]uit [a]: a
Add item: Love Actually
1: Love Actually
[A]dd [D]elete [S]ave [Q]uit [a]: a
Add item: About a Boy
1: About a Boy
2: Love Actually
[A]dd [D]elete [S]ave [Q]uit [a]: a
Add item: Alien
1: About a Boy
2: Alien
3: Love Actually
[A]dd [D]elete [S]ave [Q]uit [a]: k
ERROR: invalid choice--enter one of 'AaDdSsQq'
Press Enter to continue...
[A]dd [D]elete [S]ave [Q]uit [a]: d
Delete item number (or 0 to cancel): 2
1: About a Boy
2: Love Actually
[A]dd [D]elete [S]ave [Q]uit [a]: s
Saved 2 items to movies.lst
Press Enter to continue...
1: About a Boy
2: Love Actually
[A]dd [D]elete [Q]uit [a]: a
Add item: Four Weddings and a Funeral
1: About a Boy
2: Four Weddings and a Funeral
3: Love Actually
```

```
[A]dd [D]elete [S]ave [Q]uit [a]: q  
Save unsaved changes (y/n) [y]:  
Saved 3 items to movies.lst
```

Keep the `main()` function fairly small (less than 30 lines) and use it to provide the program's main loop. Write a function to get the new or existing filename (and in the latter case to load the items), and a function to present the options and get the user's choice of option. Also write functions to add an item, delete an item, print a list (of either items or filenames), load the list, and save the list.

Code for `get_string()` and `get_integer()` functions is provided in this week's folder in b/b.

Keep the items in case-insensitive alphabetical order, and keep track of whether the list is "dirty" (has unsaved changes).

Offer the "Save" option only if the list is dirty and ask the user whether they want to save unsaved changes when they quit only if the list is dirty.

Adding or deleting an item will make the list dirty; saving the list will make it clean again.