Alejandro Wolf-Yadlin

Professor Randal Root

IT FDN 100 A

10/31/2016

Assignment 105

In this assignment we worked with lists and dictionaries to make a to do list of household tasks and their priorities. The list can be modified by any user by adding or deleting tasks. When the user is done, the list is saved to ToDo.txt. We did this in 4 steps:

1. **We wrote the script “todolist.py” to perform this task. The script is pasted below:**

*# Assignment 5*

*# By Alejandro Wolf-Yadlin, October 31, 2016*

*# This program manages a to do list*

*# The list contains two columns of data ("Task" and "Priority"*

*# The lsts will be managed suing a python dictionary*

todolist = {} *# Define empty dictionary "todolist"*

*# Open ToDo.txt to read as a variable name "todo"*

**with** open("ToDo.txt") **as** todo: *# will close todo when loop is done*

**for** line **in** todo:

line =line.strip() *# will remove '\n' at end of new line*

(task, priority) = line.split(',') *# Separate task and priority by comma*

Task=task *# Define task in dictionary*

todolist[Task] = priority *# Assign pritority to task in dicitionary*

todotable = [] *# Define empty dictionary "todolist"*

**for** key, value **in** todolist.items(): *# Trsansfer keys and values from dictionary to list*

todotable.append([key,value])

*# This section prints the starting state of the to do list*

*# as it was read from ToDo.txt, with headers 'Task' and 'Priority'*

print('Your current Tasks and Prioriteis are:')

print('Task ', 'Priority')

**for** line **in** todotable:

print(line[0],' ',line[1])

*# This section provides the user 3 options:*

*# add task to the to do list*

*# remove tasks from the to do list*

*# Save the todo list to ToDo.txt*

choice = 0 *# Define dummy variable choice*

conditions = ('1', '2', '3') *# Define valid answers for user to input*

*# The while loop below will keep running unless the user chooses to save his/her to do list*

**while** choice != 3: *# choice = 3 means save list as is now*

print ("**\n**Choose 1 to Add task")

print ("Choose 2 to Remove task")

print ("Choose 3 to Save all tasks to the Todo.txt file and exit!")

choice = input('What would you like to do?: ')

*# if the user inputs an invalid choice, ask them to input valid one*

**if** choice **not** **in** conditions:

print('**\n**Please select choice 1, 2 or 3: ')

**else**:

*# convert choice from str to integer so it can be compared to numerical values*

choice = int(choice)

*# if the user chooses to add a task append it at end of list and dictionary*

**if** choice == 1:

nt = str(input("**\n**What is your new task?: "))

np = str(input("What is its priority?: "))

todotable.append([nt,np]) *# list*

todolist[nt]=np *#dictionary*

print('**\n**your new task:', nt,'has been added to your to do list with priority', np)

*# if the user choses to remove a task*

**elif** choice == 2:

tr = str(input("**\n**What task do you want to remove?: "))

removed = 0 *# Define dummy variable removed*

**for** t **in** todotable: *# iterate through the table lines*

**if** tr **in** t[0]: *# if the task is in any line*

todotable.remove(t) *# remove that line*

print(tr,' has been removed**\n**')

removed = 1

**if** removed != 1:

print('**\n**Your choice is not available, lets start again: ')

todotable.sort(key=**lambda** x:x[0]) *# Sort the list by task alphabetically*

todolist =dict(todotable) *# Save the list as dictionary*

*# Print to screen the*

print('**\n**Your modified Tasks and Prioriteis are:**\n**')

print('Task ', 'Priority')

myfile = open("ToDo.txt", "w") *# create/open text file, with writing privileges*

**for** line **in** todotable:

print(line[0],' ',line[1]) *# print the table to screen*

task = ','.join(str(x) **for** x **in** line) *# define each line to be printed to file*

myfile.write(task + '**\n**')

myfile.close()

input('Your to do list has been saved, please press any key to exit')

**The key functions and commands learned here are:**

* While: this command allow us to repeat a set of code while a condition holds: *while x < 5*
* For loops: this command allow us to iterate through a variable, table, tuple string as long as we are within range or dimension: *for i in range x or for I in range(0,n)*
* If/elif/else: this command allow us to anticipate different scenarios an provide adequate code according to ouropearting conditions: *if a do b, elif c do d else do e*.
* Manipualtion of list and dictionaries: we learned to build to assign new lines to lists and values to dictionary keys, for example:

list.append([nt,np]) appends [nt,np] to a list of lists

list.remove(row) will remove row from list

dict[nt]=np gives the value np to the key nt

* Learn the mini function lambda

list.sort(key=lambda x:x[0]) in thi case lambda calls the sort function to sort list according to its first column

* We learned to extract all elements of a list into string to write the into file:

for line in list:

iterate through each sublist in list

task = ','.join(str(x) for x in line)

find all string in the sublist and join them ny ‘,’

myfile.write(task + '\n')

write the strings to file and move to next line

1. **Verified simple code worked by using same protocol as in assignment 101:**
   1. Open windows command window and run program from there:

Write code (verbatim):

start python.exe c:\\_PythonClass\todolist.py

* 1. Open c:\\_PythonClass folder and run program from there:
     1. Double click todolist.py file
     2. Right click edit with IDLE & choose RUN Module (F5 key)

1. **Created Batch file to run basic\_operations.py, using same protocol as in assignment 101**
   1. Open Notepad ++
   2. Write code (verbatim):

cd C:\Python32

start python.exe c:\\_PythonClass\todolist.py

pause

* 1. Save text file as Runtodolist.bat
  2. Test batch file by going to c:\\_PythonClass and double click on Runtodolist.bat
  3. From command line type c:\\_PythonClass\Runtodolist.bat

1. **We opened the file by double clicking on ToDo.txt and confirmed that our input item and value was saved**

**Another way to do it would have been with the following code:**

my\_file = open("ToDo.txt", "r")

file\_contents = my\_file.read()

print(file\_contents)

The first instruction opens the file ToDo.txt for reading. The second line reads all contents of the file to the variable file\_contents and finally the third line prints those file contents to the screen.