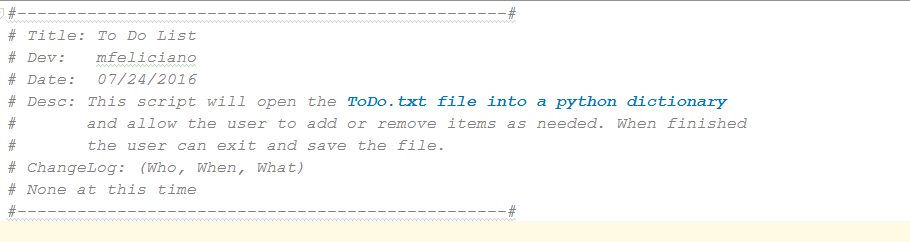
Matthew Feliciano

Instructor**:** Randal Root

IT FDN 100 A

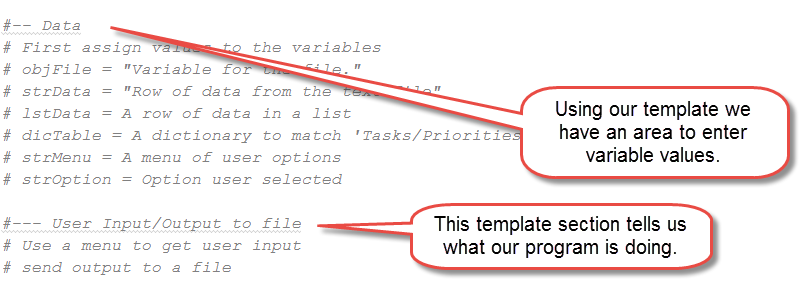
25 July 2016

1. Creation steps
   1. Open PyCharm
   2. Open new Python file
   3. Create header – This tells you or future programmers what the code is doing and who wrote it. (Note we are now using a standardized template)



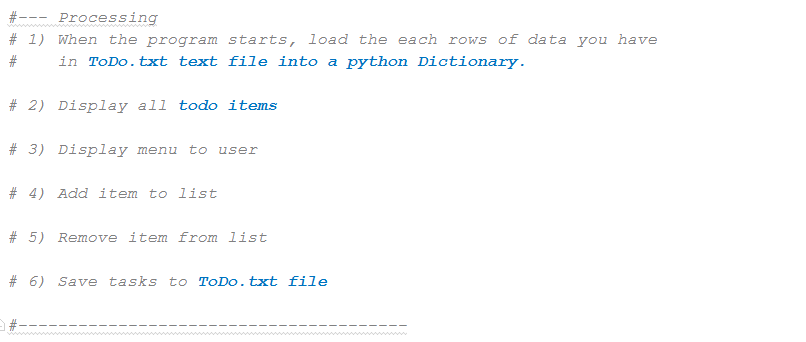
As with headers, templates, are always a best practice no matter what type of code you are writing. In addition to helping future programers, and yourself, to understand what the code is attempting to accomplish. The template provides a section in the header to document changes. Hopefully this will encourage future coders to use it. Another advantage to using a template is that it keeps all code throughout the organisation consistant. It can also hold lines of code that are required of all scripts.

1. Next we need to assign values to our variables

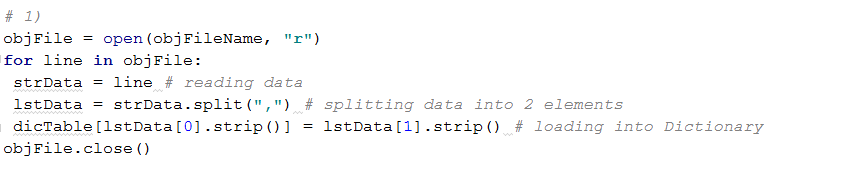


Here we are defining variables to be used later in our program. We are also discribing the how the user will input data and what the program will do with the output.

1. The Processing section of the template lets us know what the program is doing.



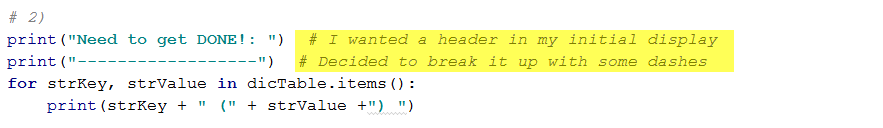
1. Open the ToDo.txt file



Here we are opening the file and loading all of the data currently in the file into a Dictionary.

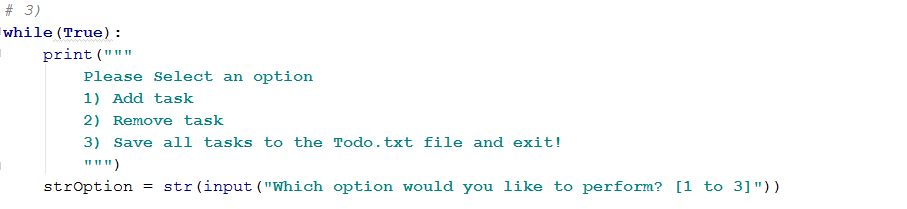
Note – the numbers here correspond with numbers in our Processing section of the template. Personally, if I were to do it again, I would add the comments as well. If there were even hundreds of lines of code it would not be fun to have to scroll back to the top to figure out what a block is doing.

1. Now we display what is currently in the file to the user.

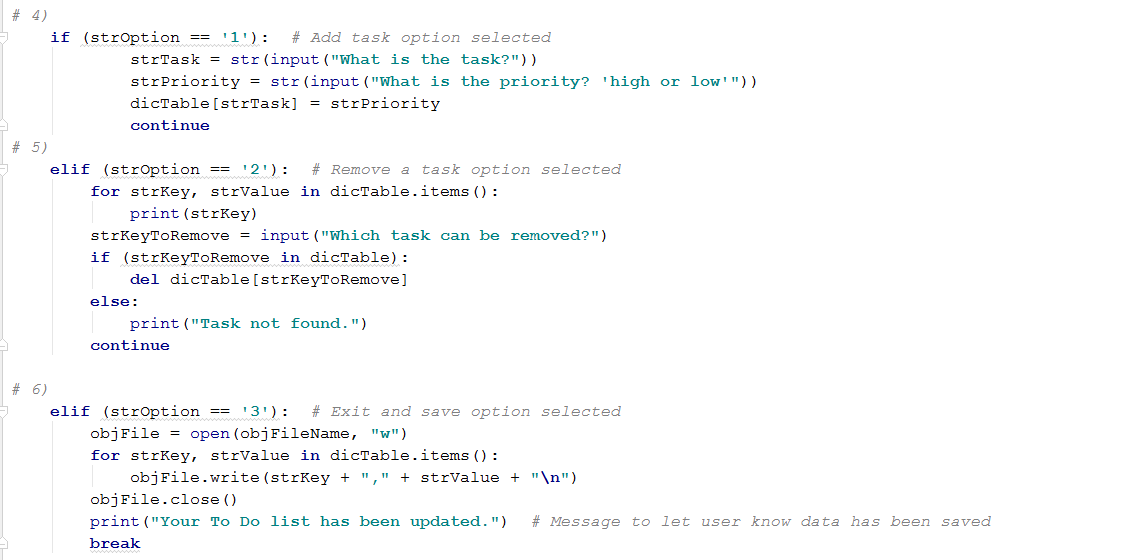


I decided that I wanted to add a header to my initial list to the user. There may be a better way of doing this but it works for now.

1. Here we are giving our user options to select to update the file as part of our WHILE loop.



1. This next section tells the program what actions to take depending on the user’s selection.

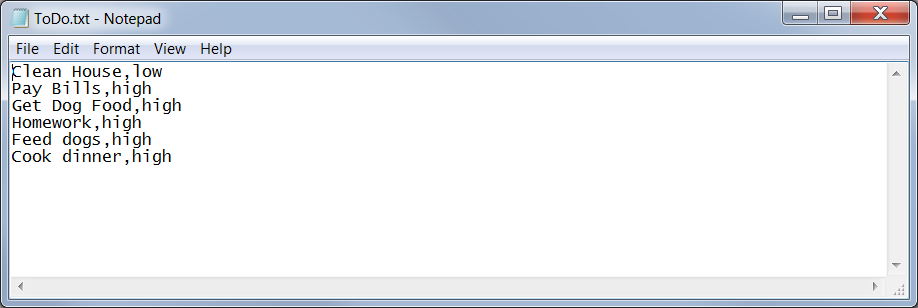


This loop is asking the user to select an option (strOption) and enter a new task (strTask) and the priority for the task (strPriority).

1. We then continue to loop, displaying the menu options, allowing the user to add and remove tasks, until they select option #3.
2. When the user selects option #3 the program will:

* Open the ToDo.txt file
* Write back the data from our Dictionary
* Close the file
* Print a message to the user saying the data was saved

1. The file contents will look something like this.



1. Python code
2. *#-------------------------------------------------#  
   # Title: To Do List  
   # Dev: mfeliciano  
   # Date: 07/24/2016  
   # Desc: This script will open the* ***ToDo.txt file into a python dictionary****# and allow the user to add or remove items as needed. When finished  
   # the user can exit and save the file.  
   # ChangeLog: (Who, When, What)  
   # None at this time  
   #-------------------------------------------------#  
     
   #-- Data  
   # First assign values to the variables  
   # objFile = "Variable for the file."  
   # strData = "Row of data from the text file"  
   # lstData = A row of data in a list  
   # dicTable = A dictionary to match 'Tasks/Priorities'  
   # strMenu = A menu of user options  
   # strOption = Option user selected  
     
   #--- User Input/Output to file  
   # Use a menu to get user input  
   # send output to a file  
     
   #--- Processing  
   # 1) When the program starts, load the each rows of data you have  
   # in* ***ToDo.txt text file into a python Dictionary.****# 2) Display all* ***todo items****# 3) Display menu to user  
     
   # 4) Add item to list  
     
   # 5) Remove item from list  
     
   # 6) Save tasks to* ***ToDo.txt file****#---------------------------------------*objFileName = **"C:\\Users\\feliciam\\Documents\\\_PythonClass\\Module05\\ToDo.txt"**strData = **""**dicTable = {}  
     
   *# 1)*objFile = open(objFileName, **"r"**)  
   **for** line **in** objFile:  
    strData = line *# reading data* lstData = strData.split(**","**) *# splitting data into 2 elements* dicTable[lstData[0].strip()] = lstData[1].strip() *# loading into Dictionary*objFile.close()  
     
   *# 2)*print(**"Need to get DONE!: "**) *# I wanted a header in my initial display*print(**"------------------"**) *# Decided to break it up with some dashes***for** strKey, strValue **in** dicTable.items():  
    print(strKey + **" ("** + strValue +**") "**)  
     
   *# 3)***while**(**True**):  
    print(**"""  
    Please Select an option  
    1) Add task  
    2) Remove task  
    3) Save all tasks to the Todo.txt file and exit!  
    """**)  
    strOption = str(input(**"Which option would you like to perform? [1 to 3]"**))  
     
   *# 4)* **if** (strOption == **'1'**): *# Add task option selected* strTask = str(input(**"What is the task?"**))  
    strPriority = str(input(**"What is the priority? 'high or low'"**))  
    dicTable[strTask] = strPriority  
    **continue***# 5)* **elif** (strOption == **'2'**): *# Remove a task option selected* **for** strKey, strValue **in** dicTable.items():  
    print(strKey)  
    strKeyToRemove = input(**"Which task can be removed?"**)  
    **if** (strKeyToRemove **in** dicTable):  
    **del** dicTable[strKeyToRemove]  
    **else**:  
    print(**"Task not found."**)  
    **continue***# 6)* **elif** (strOption == **'3'**): *# Exit and save option selected* objFile = open(objFileName, **"w"**)  
    **for** strKey, strValue **in** dicTable.items():  
    objFile.write(strKey + **","** + strValue + **"\n"**)  
    objFile.close()  
    print(**"Your To Do list has been updated."**) *# Message to let user know data has been saved* **break**