



LAB 2 – Writing a Complete Script in Python

Part 1 – Working with date and time types in Python

For this exercise, carefully read about the date and time types in Python from the following link:

<https://docs.python.org/3/library/datetime.html>

Tasks:

- Write a Python program that prompts the user to enter a date in the format "MM/DD/YYYY", and then converts it to a datetime object.
- Write a Python function that takes a datetime object and formats it as a string in the format "YYYY-MM-DDTHH:MM:SSZ".
- Write a Python function that takes a datetime object and returns the date and time as separate strings.
- Write a Python program that prompts the user to enter two dates in the format "MM/DD/YYYY", and calculates the number of days between them using the datetime library.
- Write a Python function that takes a datetime object and adds a specified number of days to it.

Part 2: Write a complete Python script, with comments, to do the following:

- Open a text file called "*catalog.txt*", attached with this lab, for reading. The file contains the items available in a fitness studio, the items categories/classes, and their quantities.
- Define a list of strings called *fit_items*. The list should contain at least 10 strings and each string represent a specific fitness item, e.g., treadmill, lifting bars, weights, etc.
- Loop over each element in *fit_items* and check if that element matches any of the products in the file.
Hint: use the function `readline()` to read a new line from the file and compare that line with the elements in the list of strings.
- If there is a match, save the category and the quantity corresponding to that item in some variables.
- Create a dict *d1* with entries *item:category* where *item* (**key**) is the item (string) found in *catalog.txt* and *category* (**value**) is the category of that item. Add the item and its category to *d1* as *{item:category}*. Create another dict *d2* with entries *item:quantity* and add the item found and its quantity to *d2*.

Hint: use the function `update()` on *d1* and *d2* to add the item found and its category and quantity to the dictionaries.

- f. Next the program should ask the user to enter a string *s*, **representing a fitness item**, as an input and retrieve the *category* of *s* from **d1** and the *quantity* from **d2** .
- After displaying the category and quantity corresponding to item *s*, the program asks if the user would like to do another search with (**yes/no**) options.
 - If the user enters **yes**, another category and quantity retrieval should be done for another item.
 - If the answer is **no**, the program should exit.
- g. If the item's name entered by the user does not correspond to a valid key, the program should catch an exception. When the exception occurs, display an appropriate error message then prompt the user to input another item's name.

Part 3: Answer the following questions:

- a. Assume that you have a list named L, e.g., L = [19, 52, 87, 2, 8, 11, 18, 22], write a Python script to count the number of odd numbers in L?
- b. Assume that you have a list named L, e.g., L = [38, 5, 7, 2, 8, 112, 18, 400], write a Python script to calculate the average of all even numbers in L?
- c. Assume that you have a list named L, e.g., L = [38, 5, 7, 2, 8, 112, 18, 400], write a Python script to find the largest and the smallest number in L?