Info8025 – IT Automation

Assignment 2

Requirements

This assignment consists of **Task 1** and **Task 2**. Each Task is to be completed in a separate script. Task 2 builds on top of Task 1.

To see how the script for each of the two Tasks is expected to work when executed**, refer to the respective screen captures** included further down in the document.

TASK 1

Write a Python script, ***Assign2_yourStudentNum_Task1.py***, that checks a directory for text files (files with .txt extension) and prints out to the console the list of the file names retrieved.

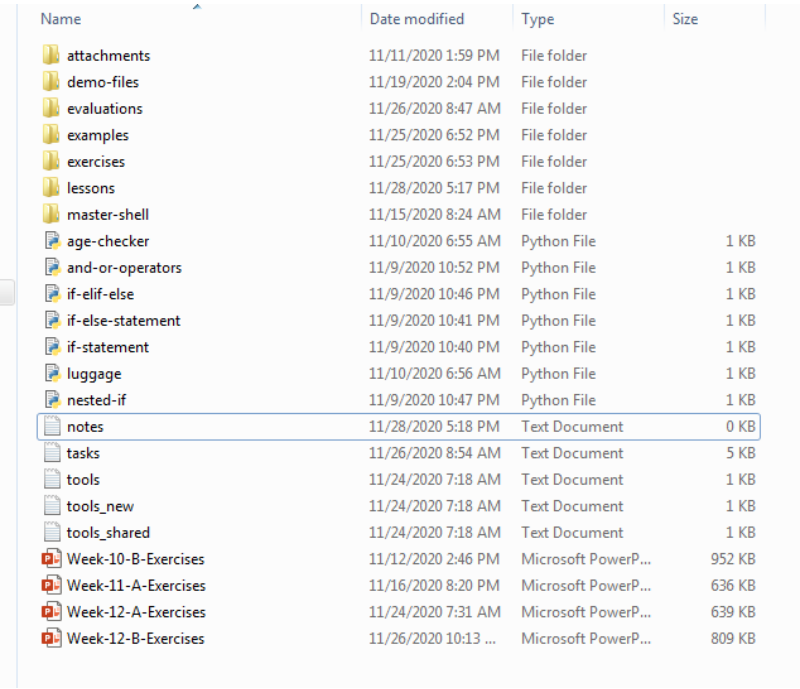
Here are the details:

- In your script, define a function that will check a directory of your choice and return the list of the text files found in it.

  Notes:
  - Use a variable to store the name of the directory to be searched.
  - Use another variable to store the absolute path to the parent directory of the directory to be searched.
  - **Use os.path.join()** to concatenate the above two values, to get the full (absolute) path to the directory to be searched.

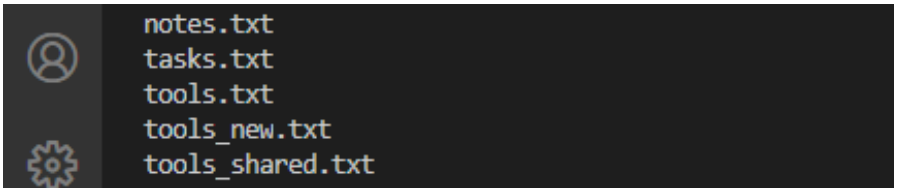
- Add the code that will call the function and use the function's return value to print to the console the names of the text files found.

The structure and contents of the searched directory on which both Task 1 and Task 2 Output (shown below) are based:



Name	Date modified	Type	Size
attachments	11/11/2020 1:59 PM	File folder	
demo-files	11/19/2020 2:04 PM	File folder	
evaluations	11/26/2020 8:47 AM	File folder	
examples	11/25/2020 6:52 PM	File folder	
exercises	11/25/2020 6:53 PM	File folder	
lessons	11/28/2020 5:17 PM	File folder	
master-shell	11/15/2020 8:24 AM	File folder	
age-checker	11/10/2020 6:55 AM	Python File	1 KB
and-or-operators	11/9/2020 10:52 PM	Python File	1 KB
if-elif-else	11/9/2020 10:46 PM	Python File	1 KB
if-else-statement	11/9/2020 10:41 PM	Python File	1 KB
if-statement	11/9/2020 10:40 PM	Python File	1 KB
luggage	11/10/2020 6:56 AM	Python File	1 KB
nested-if	11/9/2020 10:47 PM	Python File	1 KB
notes	11/28/2020 5:18 PM	Text Document	0 KB
tasks	11/26/2020 8:54 AM	Text Document	5 KB
tools	11/24/2020 7:18 AM	Text Document	1 KB
tools_new	11/24/2020 7:18 AM	Text Document	1 KB
tools_shared	11/24/2020 7:18 AM	Text Document	1 KB
Week-10-B-Exercises	11/12/2020 2:46 PM	Microsoft PowerP...	952 KB
Week-11-A-Exercises	11/16/2020 8:20 PM	Microsoft PowerP...	636 KB
Week-12-A-Exercises	11/24/2020 7:31 AM	Microsoft PowerP...	639 KB
Week-12-B-Exercises	11/26/2020 10:13 ...	Microsoft PowerP...	809 KB

Task 1, Program output: The program found and listed the names of all text files found in the default directory specified in the program.



```
notes.txt
tasks.txt
tools.txt
tools_new.txt
tools_shared.txt
```

TASK 2

In this task you are asked to modify and improve the program created in Task 1, by making it more flexible and by expanding its logic.

You are to write a script which, when executed, will search a specified directory for files of a specific type, which names start with a specific string. The directory, file type and the string to look for in a file name are provided by the user when the script runs. The search results are stored into a newly created text file and a message is displayed to the console.

The output of the program should match the output shown in the screen captures, below.

1. **Step 1: Important: Start by making a copy of your completed script from Task1.** Name this new file ***Assign2_yourStudentNum_Task2.py***.

2. **Step 2: Modify the code** in *Assign2_yourStudentNum_Task2.py* script.

   Here are the **details on how the program is expected to work:**

   - Ask the user to enter the following information:
     - The name of the directory to search through
       - Note: this should be one of the directories that exist in the parent directory stored in a variable (in Task 1)
     - An extension that will indicate the type of file to search for in the directory
     - The string to look for a match in file names
   - Accept user's input and pass each piece of information provided as an argument in the call to the function.
   - Accept the return value from the function (ie. a list of file names), and:
     - If at least one matching file is found:
       - Create a new text file, named fileNames.txt, in the searched directory.
       - Write the list of file names returned by the function to the created file. Note: the file names should be written **one file name per line** (see the screen shot)
     - Print a message to the console that will display the following:
       - The number of matching files found
       - The full path to the file where the results where stored – if applicable (ie. if at least 1 matching file is found and the text file is created.)
   - Modify the function you have created earlier so that it:

- **Accepts 3 parameters**: directory name, an extension, and a string to look for in file names

- Uses the values passed to it as parameters to perform the functionality for which it was created, that is to search **the specified directory** for the files with **the specified extension** that have their **name start** with **the specified string**.

`

Notes: Make sure the **search is not case sensitive**:

a) In regards to the file extension; For ex: whether the user enters .TXT, .txt, or .Txt – these should all be treated as same and valid, ie. any file with a .txt extension will be a match.

b) In regards to the condition that the file name starts with the specified string. For ex. if user enters 'wee' as the search string, a match is found if the file name starts with 'Wee', 'wee', or WEE', etc.

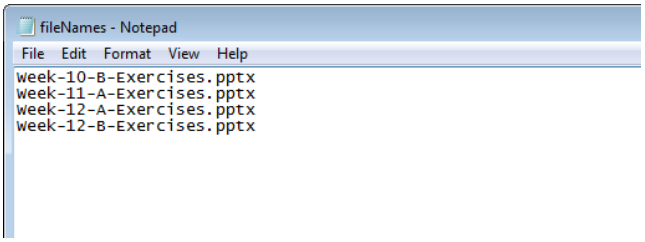
- **Uses a Regular Expression** when checking if a file name starts with the string provided by user.
- **Uses os.path.join()** whenever concatenating parts of a path into a full path.
- **Returns a list of files** that meet the conditions above (extension and name)

- **Make sure that your final Task 2 code is clean**.
  After changing your program in Task 2, you might end up with some lines of code from Task 1 that you don't need any more (perhaps some variables, etc). Make sure that you remove any unused, left-over code from Task 1.

Task 2, Program Output: Console

```
Enter the name of the directory to search:
python
Enter the type of files you want to search for, by indicating the file extension to use:
.PPTX
Enter the starting few characters of the file name:
WEE
4 matching files found.The list of file names retrieved can be found in fileNames.txt file: D:\bilja\
conestoga\2019-2020\COURSES\F20-INFO8025-IT-Automation\fileNames.txt
```

Task 2, Program Output: Content of fileNames.txt file



Submission

You are required to submit the following **3 things** to the **Assignment 2 Dropbox** on eConestoga:

1. **Your 2 script files**:
   - *Assign2_yourStudentNum_Task1.py*
   - *Assign2_yourStudentNum_Task2.py*

2. **A URL to a screen recording of you doing a walk-thru on your code** by reflecting what each of the individual lines and sections of the code are accomplishing.

   Notes re: the recording:
   - Create a single, up to 8 minutes long video.
   - Upload the recording to YouTube.
   - Make sure the video is made public so that I can view it.
   - Submit the URL to your video to the Assignment 4 Dropbox on eConestoga.

Assessment

- You will be graded by the rubric found on eConestoga, under Course Tools -> Rubrics.
- This is an independent work assignment. You may not work with others while working on this assignment or collaborate or share your answers, code or anything else.
- You may use the book, and the course material found on eConestoga.
- You may use the web, for generalized Python coding practices research.
- You may not use code copied from other people, the web or any other source.