

Assignment 9 – Language Translator

Goals

- File processing (writing and reading)
- Working with CSV files
- Practice with functions, lists, and loops

Background

- You are provided a CSV (comma-separated value) file with words in English and 14 other languages.
 - Each row represents one line in a table, and commas separate each column.
 - The first line in this CSV file represents the header, and each subsequent row represent the data in each column.
 - If there is no translation for a word, then there is a dash (-).
 - **CSV File (Example)**
English,Danish,Dutch,Finnish,French,German
animal,dyr,dierlijk,eläin,bête,tier
cat,kat,kattekop,kissa,chat,katze
dog,hund,hond,koira,clebs,hund
snake,slange,slang,-,guivre,schlange

Data Table (Example)

English	Danish	Dutch	Finnish	French	German
animal	dyr	dierlijk	eläin	bête	tier
cat	kat	kattekop	kissa	chat	katze
dog	hund	hond	koira	clebs	hund
snake	slange	slang	-	guivre	schlange

Requirements

- Create a new Python file. It must begin with comments in the following format (replace the name and email with your actual information and write text for the description):
Name, USC email
ITP 115, Spring 2020
Assignment 9
Description:
Describe what this program does.
- All operations must be done using techniques discussed in class (including functions).
You may NOT import the CSV or other modules.

- Your program must complete the following tasks:
 - Read in a CSV file with words in 15 languages to create a list of words in English.
 - Ask the user to select a language and read the CSV file to create a list of words in that language.
 - Ask the user for a word, translate the word, write it an output file, and repeat until the user is done.
- Notes:
 - You are not allowed to create any global variables.
 - You need to get the names of the languages from the CSV file. You cannot hard-code any of the names of the languages except when assigned "English" as the default language.
 - A dash (-) means that there is not a translation for that language.

Functions to implement

- **getLanguages(fileName)**
 - Parameter: fileName is a string containing the name of a CSV file to read from and it has a default value of "languages.csv"
 - Return value: a list of strings representing the languages in the header row
 - Open the CSV file and get the header row with the languages and put it into a list.
 - Close the file and return the list.
- **getSecondLanguage(langList)**
 - Parameter: langList is a list of the languages
 - Return value: a string for the second language
 - Display to the user the languages that are available for translation.
 - Get input from the user for the second language. They must enter in a valid language. The user's input is not case sensitive.
 - Return the language.
- **readFile(langList, langStr, fileName)**
 - Parameter 1: langList is a list of the languages
 - Parameter 2: langStr is a string of containing the name of a language and it has a default value of "English"
 - Parameter 3: fileName is a string containing the name of a CSV file to read from and it has a default value of "languages.csv"
 - Return value: a list of words in the language identified by the langStr parameter
 - Open the CSV file and read the header row to skip it.
 - Use the langList and langStr parameters to determine which column of data to save.

- Loop through the rest of the file to create a list of words.
- Close the file and return the list.
- **createResultsFile(language, resultsFile)**
 - Parameter 1: language is a string containing the name of the second language
 - Parameter 2: resultsFile is a string containing the name of the results file
 - Return value: none
 - Open the results text file such that if there is an existing file, it will be overwritten.
 - Write text to the file stating the second language.
Example text to file: Words translated from English to German
 - Close the file.
- **translateWords(englishList, secondList, resultsFile)**
 - Parameter 1: englishList is a list of words in English
 - Parameter 2: secondList is a list of words in the second language
 - Parameter 3: resultsFile is a string containing the name of the text file
 - Return value: none
 - Open the results file in order to append text into it.
 - Ask the user to enter an English word to translate.
 - If the word is not in the English list, then display a message to the user.
Example message: turtle is not in the English list
 - Translate the word. If there is a translation, then display a message to the user and write the word and its translation to the file.
Example message: rabbit is translated to kani
Example text to file: rabbit = kani
 - If there is no translation (i.e., the value is "-"), then display a message to the user.
 - Ask the user if they want to translate another word and repeat until they answer "n" (case insensitive).
 - Close the file.
- **main()**
 - Call the function you created to get the list of languages
 - Call the appropriate function to read the CSV file for the English words.
 - Call the appropriate function to get the second language.
 - Call the appropriate function to read the CSV file for that language.
 - Ask the user to enter a name for the results file. Use the second language with ".txt" as a default file name. If the second language is Italian, then the default file name is "Italian.txt". If the user enters only the enter/return key, use the default name.

- Call the appropriate function to write a message to the results file.
Example text to file: Words translated from English to German
- Call the appropriate function to allow the user to enter word, translate those words, and save to the text file.

Sample Output 1 – Screen Output

Language Translator

Translate English words to one of the following languages:

Danish Dutch Finnish French German Indonesian Italian

Japanese Latin Norwegian Portuguese Spanish Swahili Swedish

Enter a language: chinese

This program does not support Chinese

Enter a language: finnish

Enter a name for the results file (return key for Finnish.txt):

Enter a word to translate: rabbit

rabbit is translated to kani

Another word (y or n)? q

Another word (y or n)? y

Enter a word to translate: snake

snake did not have a translation.

Another word (y or n)? y

Enter a word to translate: turtle

turtle is not in the English list.

Another word (y or n)? y

Enter a word to translate: bird

bird is translated to lintu

Another word (y or n)? n

Translated words have been saved to Finnish.txt

Sample Output 1 – Finnish.txt File

Words translated from English to Finnish

rabbit = kani

bird = lintu

Sample Output 2 – Screen Output

Language Translator

Translate English words to one of the following languages:

Danish Dutch Finnish French German Indonesian Italian

Japanese Latin Norwegian Portuguese Spanish Swahili Swedish

Enter a language: spain

This program does not support Spain

Enter a language: Swahili

Enter a name for the results file (return key for Swahili.txt):
results.txt

Enter a word to translate: earth

earth is translated to kiwanja

Another word (y or n)? a

Another word (y or n)? y

Enter a word to translate: peace

peace did not have a translation.

Another word (y or n)? y

Enter a word to translate: school

school is translated to shule

Another word (y or n)? y

Enter a word to translate: city

city is translated to mji

Another word (y or n)? y

Enter a word to translate: teacher

teacher is not in the English list.

Another word (y or n)? n

Translated words have been saved to results.txt

Sample Output 2 – results.txt File

Words translated from English to Swahili

earth = kiwanja

school = shule

city = mji

Deliverables and Submission Instructions

- Create a folder on your computer called **ITP115_A9_LastName_FirstName** (replace *LastName* with your last/family name and *FirstName* with your first name).
- Inside the folder, put your python source code and the CSV file.
- Compress the folder (make a zip file). This cannot be done within PyCharm. Find the folder on your computer and compress it.
 - a. Windows:
 1. Using File Explorer, select your folder
 2. Right click
 3. Send to ->
 4. Compressed (zipped) folder
 - b. Mac OSX:
 1. Using Finder, select your folder
 2. Right click
 3. Compress "*FolderName*"
- Upload the zip file to your Blackboard section:
 1. On Blackboard, click on the Assignments item in the course menu on the left.
 2. Click on the specific item for this assignment (starts with A and a number).
 3. Click on the Browse My Computer button and select your zip file.
 4. Click the Submit button.

Grading

Item	Points
getLanguages()	4
getSecondLanguages()	4
readFile()	8
writeToFile()	2
translateWords()	8
main()	4
Total*	30

** Points will be deducted for poor code style, lack of error checking, improper submission.*