Introduction to Python: Assignment 2

Handed out:

Due: 11:59 PM, Monday, February 25, 2019

In this assignment you are going to build a program to analyze song lyrics.

Submission: You should save your code as assignment2.py
The first line of code should be a comment with your full name, and student number. [failure to comply would cause no credit at all]

Submitted code must be syntax clean.

Don't forget to include comments. Grades are partially based on readability.

Collaboration: You may work with other students; however, each student must write up and hand in his or her assignment separately. Be sure to indicate with whom you have worked in a comment at the start of each file.

Instructions:

In this assignment you are going to write a program that analyzes song lyrics.

You will start by loading the song lyrics as a list. Then you will write a function that creates a frequency dictionary; Simply every word is a key and values are the frequency of the word.

You will then develop three functions for analysis.

1. Getting you set up:
    a. Download the two files on the assignment page on canvas: assignment2.py and helper2.py
    b. Save both files to your current Spyder directory.
    c. Run helper2.py (this will generate a list **she_loves_you** that contains the lyrics of the Beatles song “She loves you”.) Check your variable explorer to make sure the list is loaded.
2. Writing the functions:
    a. In the assignment2.py file you will find all the function declarations that you need to develop. The docstring is ready for you, all you need to do is to write the code.
3. Testing your functions:
    a. Try to test each function separately before integrating everything together.

Sample run:

```
songDict=lyrics_to_frequency(she_loves_you)
#finding the most common word
topWord=most_common_word(songDict)
print(topWord)

#finding words that occur at least 5 times
mostWords=words_with_minimum_freq(songDict,5)
print(mostWords)
```

```
(['you'], 36)

[(['you'], 36), (['yeah'], 28), (['she'], 20), (['loves'], 13), (['know'], 11), (['be'], 10), (['and'], 8),
(['that', 'should', 'glad'], 7), (['love'], 5)]
```

Ungraded part:

Here are some ideas to make this program more complete:

- In the graded section, we used a ready-made list of strings; How about writing a function that takes in a string of a song (you can copy the string from any website) and converts it to a list of strings—This way we can analyze more and more songs.
- Cross analysis of two different songs:
    - Write a function that takes two lists of two different songs and finds the words that occurred the most in both songs combined.
    - Write a function that finds words that were used in one song but not the other one.