INFO SHEET BEFORE STARTING TO PROJECT

Dictionaries :

In the lecture, we saw lists in the collection data types. Dictionaries are just like lists, but instead of the index, they have keys to access to the values**. It will be wiser to use dictionary type in this Project.**

So let’s learn dictionary by comparing it to list:

```
>>> List_sample = ["a", "b", "favprogram", [1, "world",100]]
```

0	1	2	3
"a"	"b"	"favprogram"	[1,"world",100]
-4	-3	-2	-1

- If we want to add a new member to the list, we need to use append or insert functions.
- If you need to access to “favprogram”, List_sample[2] or List_sample [-2] will return the value.
- Empty list: List_sample=[ ]

Dictionary

```
>>>phoneBook  = { "Smith"  :  3253, "Johnson"  :  3938, "Brown"  :  1443,  "Miller": 9388 }
```

KEY	VALUE
Smith	3253
Johnson	3938
Brown	1443
Miller	9388

- Access a value using a key (but not vice versa!)
  ```
  >>> phoneBook ["Smith"]
  3253
  ```
- A dictionary can’t contain multiple items with the same key. (HINT: In this project, every word you have will be a key. And how many times they occur will be the value of that key.)

- Values don’t have to be unique.
- Empty dictionary: **phoneBook = dict() or phoneBook = { }**
- If you want to add a new item to dictionary, unlike lists you can use direct assignments. (No need for append or insert)

```
>>> phoneBook ["MaryPoppins"] = 3938
>>> phoneBook
{ "Smith"   :  3253, "Johnson"   :  3938, "Brown"   :   1443,   "Miller": 9388,
"MaryPoppins": 3938 }
```

- dict.keys() - returns a view of all the keys of a dictionary.
- dict.values() - returns a view of all the values of a dictionary.
- dict.items() - returns a view of all the items in a dictionary (this returns pairs in dictionary). Please take the piece of code below and see what it prints.

```
phoneBook  =  {'Smith':  3253, 'Johnson':  3938,
'Brown':  1443, 'Miller':  9388, 'MarryPoppins':  3938 }

persons  = phoneBook.keys()
for  person  in  persons:
     print(person, ">>", phoneBook[person])

entries  = phoneBook.items()
for  entry  in  entries:
     print(entry[0],  ">>",  entry[1])
```

PROJECT TEXT

Objectives:

1. Read & process input data.
2. Use conditionals, loops, variables and simple data structures (e.g. lists, dictionaries , etc.)
3. Use simple statistics.
4. Plot output data.

For this exam, you will be working in teams of two. First find your project partner from your lab sessions before attempting to solve the exam problems. If you can not find a team member, you can do the project by yourself.

You can copy the template.py and turkish_data_protection_law.txt files to trinket python3 or colab. Work on the template.py script.

In this project, your goal is to do a rudimentary characterization of an author's use of words by computing the frequency of her/his/their 10 most popular words and display the results as a histogram.

a)

During the semester, you wrote essays, or different response papers from different courses. You will analyze your word usage as different individuals and also in different subject areas. For this purpose, please pick one essay from each course.

- Origins and Consequences I
- History of Humankind

For the History of Humankind, pick Group Assignment II or III. For those assignments, you were working in groups of four. So, if you and your team member were in the same group in the History of Humankind, now one of you should pick Group Assignment II and the other Group Assignment III. You should end up with four different essays as a team.

These **four** essays (per team) will be used as input files (make sure to save the essays as plain text files with the .txt file extension).

If you are not working with a team member, work with just two essays (one essay from each course).

b) The next step is to characterize these files by some simple statistics. You will characterize the essays of each team member by the words that appear most frequently in her/his/their essays. (Please study the template.py script together with the anatomy table. There are many comments and to-do lists provided to you in .py script. The template.py works and creates an empty .png file at this point.)

ANATOMY OF THE PROGRAM

FUNCTIONS	PURPOSE
read_file	Reading your input file.
remove_stopwords	Cleaning the text from punctuations, etc…
get_sorted_freq	Heart of the program. Counts the number of occurrences of each word and sort the dictionary.
get_n_freq	Copy the 10 most frequent words and their counts to a new dictionary.
plot_hist	Plot the histogram of dictionary keys vs values.

i. **Function read_file :**
The first function is for reading your files. You don't need to modify anything inside this function.

ii. **Function remove_stopwords**:
Cleaning the text that you read in (i).
Do not count the punctuation marks, the articles (*the, a* and *an*) and the conjunctions (*for, and, or, nor, but, yet* and *so*).

a. Inside remove_stopwords, you should add punctuation marks to punctuation_list (Use google for finding all possible punctuations).
b. Inside remove_stopwords, you should add the articles and conjunctions to con_art_list.
c. After building these lists, by iterating over them with a for loop you will replace each with space. So, you don't count these stopwords, because most probably they are the most frequents in your text. The for loop for the con_art_list is given to you. Just imitate that loop for punctuation marks also (See template.py).

iii. **Function get_sorted_freq :**
To characterize an author based on a particular subject, we build a Python dictionary, one for each subject (ie. three essays) and author. From these essays, we build a dictionary of the 10 most frequently used words and their counts.

a. get_sorted_freq script is written for you.
b. You need to change the name of the dictionary to your first name.
c. Also, change the iterator in the loop to your lastname.
d. If you are working in groups, use both team members' names without any space and all lower-case letters. An example is given to you inside the script.

iv. **Function get_n_freq**:
This function is for copying the most 10 frequent words from (iii) into a different dictionary. The keys of the dictionary are your words, and their values are the count for how many times you used that word. Don't change any variable names. Just fill in the blanks in the for loop.

v. **Function plot_hist**:
This function is for plotting the histograms.
1) You need to return the plots in the return statement.
2) Additionally, the variable names which are your_surname should be changed.

vi. Make sure your program is case-insensitive (ie. counts *Origins* and *origins* as the same word) by using Python's built-in string function lower().

c) Once you count all distinct words in each essay, create and display a histogram, the x-axis showing the top-10 words and the y-axis showing the number of occurrences of those words for each essay.

d) You will have four histograms at the end. If you submit only one histogram, you cannot get full points.

e) **Advice from an old programmer**: The file **"turkish_data_protection_law.txt**" will be provided to you as your sample .txt file. You can work with that file first to check your steps. After having a code that runs without errors, you can run the same program with your own .txt files.

Good luck!

Files to Submit (Each team member should upload the same docs(!), to **KHASLEARN, LAB SESSIONS FINAL PROJECT TAB**)

1. Submit your script as **wordfreq.py**

   You have to stick to the number of lines in the template.py script. You can at most add 3 extra lines to this code.

   This file must include the names of both team members as a comment on top.

2. The four histograms as .png files.

   Name each histogram file based on this convention:
   Essay1_studentLastName-History.png
   Essay2_studentLastName-History.png
   Essay1_studentLastName-Origins.png
   Essay2_studentLastName-Origins.png

3. Also please record **a two-minute** video that explains what you did. (One video for the group is enough)

   Name the video IPython_studentLastName.mpeg

   <u>Content of the video:</u>

   - Show the essays that u picked for this project. Each student shows her/his/their essays.
   - Show the scripts. You need to show each function that you edited clearly. We gonna investigate the correctness of your codes first from here- so please show them clearly.
   - Show us your 4 plots.
   - Interpret those plots in terms of the differences of the most frequent words you used in each subject area. (2 points-Bonus)

Before you submit:

- Make sure your python program files can be compiled and run without any errors. If your submitted program file raises errors when we try to test it, you will get a zero.
- Make sure your input essays and the corresponding histogram files are unique to each team member.