Traitement Automatique du Langage
TP 1 — Python Programming

Haozhou Wang

Exercises prepared by Yves Scherrer

19.09.2019

Submit by 02.10.2019 midnight

1 Interactive interpreter

Launch the interactive interpreter and declare a list containing the following words: *how, why, however, where, never*.

Write a program (directly in the interactive interpreter) that displays, for each element of the list, a star, then the two first characters of the element, and then the whole element. The result should look like this:

```
*       ho      how
*       wh      why
*       ho      however
*       wh      where
*       ne      never
```

Modify your program so that it displays the star only when the elements start with *wh*, and print a dash (-) otherwise.

Indication: To submit this exercise, copy and paste the instructions of the interactive interpreter into a text file.

2 List comprehensions

a) Let *text1* be a sequence (list) of words. What does the following instruction do?

sum([len(w) for w in text1])

Complete the instruction such that it computes the mean length of the words in *text1*.

b) Let the list *["she", "sells", "sea", "shells", "by", "the", "sea", "shore"]* be given. Using **list comprehensions**, display:

1) all words that start with *sh*, and

2) all words that contain more than 4 characters.

3 Word frequencies

Create a script (in a file) that counts the frequency of each word in the text below:

Le poids politique de Lorient s'affirme à partir de la Révolution française et la ville gagne un rôle administratif à partir du premier Empire Les activités commerciales restent alors en retrait dans la première moitié du 19e siècle en raison des conflits fréquents mais les activités *militaires gagnent en importance*

To do so, assign the text to a variable, transform it into lower case characters, and convert it into a list of words. Then, create a dictionary where the keys are the words and the values are the frequencies. Print the words and their frequencies in alphabetical order:

```
19e        1
activités        2
administratif   1
alors    1
commerciales       1
...
```

4 An index

Create a script that takes as input a file of text data and displays the list of words contained in that file in alphabetical order, as well as the lines in which the words appear. This functionality is called **index**. Here is an example:

```
...             ...
catch           33
cause           103
causes          80
cease           64
ceased          13
ceasing         74
centre          73
certainly       39 49 74 89 100 123
chance          127
chances         43 63
...             ...
```

You may reuse parts of exercise 3. You may test your program with the file *austen.txt* available on Moodle.

5 A concordancer

A concordancer is a tool for exploring textual data. Given a corpus of text data and a word, it extracts the occurrences of this word as well as the left and right contexts in which this word appears.

For example, if given the preceding paragraph and the word *a*, it would display the following result:

```
                            a      concordancer is a to...
      a concordancer is     a      tool for exploring t...
   ... textual data. given  a      corpus of text data ...
  ...pus of text data and   a      word, it extracts th...
```

Create a concordancer in Python using object-oriented programming. You should create a class *Concordancer*, an initialization method that takes as a parameter the file name and loads it, and a method *display()* that takes as a parameter a word and displays its concordances.

You can test your program with the file *austen.txt* available on Moodle.