

List and Dictionary

Exercise 1. Hidden MESSAGE.

In the file `MESSAGE.txt` there is a mess of characters. Within this mess there is a hidden message for you. Your task will be to extract this message.

In order to get access to the content of the file, you should use the following code:

```
with open('MESSAGE.txt') as f:  
    content = f.read()
```

The above will be introduced next week. To ease your task, we decided to put those lines in the program for you and the content of the file will be passed to your function as a parameter.

Task a

Count how frequently certain symbol appear in the mess. Here comes the dictionary. The dictionary keys will be the symbols from the mess and the values will be counts of how many times each symbol appears. We start with an empty dictionary. Then for every symbol in the mess of symbols, if we have seen the symbol before, we add one to its count, and otherwise we set the count for that symbol equal to 1.

Example of resulted dictionary:

```
dict={'!': 834, '#': 920, ... }
```

Use the skeleton `HiddenMESSAGEA.py`. Write a function `dictionary(content)` which return a dictionary of symbol occurrence in the `content`. To test your application, run the program file `HiddenMESSAGEATest.py`

Task b

You should notice that some of symbols in the mess occurred only once. Your task is to arrange them in order of occurrence in the mess using `list`. Join symbols from the list into string using

```
"".join(list_of_symbols)
```

and you will get the hidden message. Let your function to return this string. Resulted message is **heterogram**, see the description at the bottom of this page.

Write a program using skeleton `HiddenMESSageB.py`. To test your application, run the program `fileHiddenMESSageBTest.py`

*HINT: Notice that python dictionaries are always unordered, it means that you can **not** use constructed dictionary to unveil the message. Probably you need to use dictionary together with the list to remember the order of occurrence*

Bonus exercises

For those of you who rapidly complete 1'st exercise and has nothing to do there is a <http://www.pythonchallenge.com/> where you have to go through programming puzzles. For each 3 solved levels you will get 1 extra point. Please report to one of TA the link to the last level you have advanced to. Note that some of the puzzles can require more knowledge than our course covers. Feel free to learn independently.

*Remark: A **heterogram** (in literature) is a word, phrase, or sentence in which no letter of the alphabet occurs more than once.*

Disclaimer: This exercise is inspired by Python Challenge level 2 puzzle <http://www.pythonchallenge.com/>