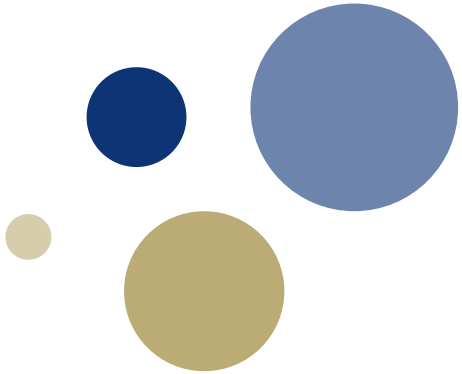

Workshop

Week 42

Warm-up

Last week we saw how we could combine the open function with a for statement to read the content in a text file line-by-line.

The text file “populations.txt” contains population numbers for several countries, where each line is one country.

Open the text file and store the population data in a **dictionary** called `records`.

Notice: the *keys* in the dictionary should be the country names, while the *values* are the actual population numbers.

Exercise 1a

In exercise 1 last week, you were asked to write a program that asked the user to supply a table of grades. The table was stored in a nested list, where each sublist contained the grades for each student.

Now, modify the solution proposal from last week, so that the table of grades is instead stored in a dictionary.

Write a program that:

- prompts the user for the number of students and tests
- for each student:
  - prompts for the name of the student
  - prompts for the score on each of the tests
  - store the name and scores in a dictionary
- display the final table of grades to the user

Notice: the *keys* in the dictionary should be the names of the student, and the *values* should be lists with the scores.

Exercise 1b

Expand the program from the previous exercise so that it also displays the average grade for each student and on each test.

Exercise 2a

In exercise 2 in last week's workshop, we wrote a program that read the text file "nurseryrhyme.txt" and counted the number of unique words in file.

We then used lists to store the words in the file. Now, we will use sets instead.

Write a program that opens the text file, stores all the words in the text in a set called `unique_words`, and daiplays the number of unique words to the user.

Exercise 2b

Make two modifications to the solution to the previous exercise:

1. Use the cleanWord function that we created in the last workshop (see exercise 2b) to clean each word before it is added to the set.

2. Place the program inside a main function.

Exercise 3a

The text file «textabbv.txt» contains a list of common abbreviations in the English language.

Write a program that opens the text file and stores the abbreviations in a dictionary called abbv_dict. The keys should be the abbreviations, while the values should be the translations.

Exercise 3b

Write a function called translateAbbrev that translates an abbreviated word.

The function should take two inputs: a string and a dictionary with abbreviations, and it should return the translated string.

Notice: the function should remove potential punctuation marks (.;,:!?) from the string before translating it.

Exercise 3c

Write a program that prompts the user for a message to translate using the dictionary with the abbreviations and the translateAbbrev function.

Implement the program in a main function.