Unit 11 Exercises – Dictionaries

Aims and objectives

- Learning about and working with dictionaries

This week's exercises

These exercises involve *dictionaries*. A dictionary is an important data structure in Python, explained in the lecture on Friday and in Chapter 12 of the book.

Task 1 – Re-read Chapter 12

Read Chapter 12. Work through the examples with the Python interpreter, and try some similar examples of your own.

Task 2 – Word frequencies

The problem is to define a function which is given a string and returns a dictionary with words (strings) as keys and numbers as values, showing how many times each word occurs in the original string. For example, when given the string:

```
"The first test of the function."
```

the function should return

```
{ "the":2, "first":1, "test":1, "of":1, "function":1 }
```

although of course the items might be in a different order. The function should ignore non-letters, and convert everything to lower case. Test the function. If you want to, you can combine it with your previous work to produce a program that displays a barchart of word frequencies in a given text. Do this work in the file *words.py*.

The function *split* from the string module will be very useful for this exercise. It separates a string into blocks, using a given string to indicate where the blocks begin and end. For example,

```
split("The first test of the function.", " ")
```

(the second argument is a space) returns

```
[ "The", "first", "test", "of", "the", "function." ]
```

Think carefully about how you are going to do this exercise, especially how you are going to convert from the original string into a list of individual words, in lower case, with all the non-letters removed. You might find it useful to develop some small functions to carry out the individual steps of this process.

Task 3 – Birthday book

The idea of this exercise is to store people's birthdays and produce reminders of birthdays that are coming up. A birthday consists of a month and a date, which can be represented by a dictionary such as

```
      { "month":"Sep", "day":17 }
```

The birthday book is a dictionary in which the keys are people’s names, and the values are birthdays, represented as dictionaries as above. The functions you have to define are described below; they should all take a birthday book as a parameter, as well as any other parameter specified below. Before you start coding, make sure you have a clear idea of how to produce the desired information.

Step 1: Working in the file *birthday.py*, write some code to set up a birthday book with several people and their birthdays, for testing purposes.

Step 2: Define a function which, given a person’s name, prints his or her birthday.

Step 3: Define a function which, given a month, prints a list of all the people who have birthdays in that month, with the dates.

Step 4: Define a function which, given a month and a date, prints a list of all the people who have birthdays within the next week, with the dates. Don’t forget that some of these birthdays might be in the next month, and if the given date is in December, some of them might be in January.