8. Sesión nº 8

Objetivos: use simple data structures such as lists, tuples, sets, and dictionaries in python; reinforcement of loops and functions; file input/output.

1. Create a file that will contain pairs of passengers and seats. Each line should contain the name of the person and then an identifier of the seat, separated by a comma, for example:

   ```
   Maria, 2A
   Manuel, 21D
   Dorotea, 14F
   Samuel, 10C
   ```

   If we have a file like this named `data.txt`, the following program will read this file and make some changes on the format, saving the result in another file named `data2.txt`:

   ```
   fin = open('data.txt','r')
   lines = fin.read().splitlines()
   fin.close()

   fout = open('data2.txt','w')
   fout.write('Reservations list:\n')
   fout.write('------------ -----\n')
   for line in lines:
           fout.write('   '+line+'\n')
   fout.close()
   ```

   Write a program that reads the file and converts it automatically into a file that will consists of two lines: the first line will contain the identifiers of the seats and the second line will contain the corresponding names.

   Optional: try to sort both lines according to the seat identifier.

2. Implement a small reservation system for an Airbus A320. The program should present a main menu where actions are selected by using single characters. For example, `r` activates the reservation module, `l` activates the list generation module, `q` quits the program.

   The menu should contain at least the following modules:

   - read a file (in the same format as given above)
   - list all occupied seats
   - add a reservation given a name and a seat number (take care, only one passenger per seat, and only available seats)
   - store the reservations list into a file
   - cancel a reservation (either by passenger name or by seat number)
   - show unoccupied seats
   - extend your program with functions you consider useful (especially for de-

bugging purposes).

In order to avoid making things too complicated, assume that the seat numbers are consecutive and that all rows have a constant number of seats.

Make good use of the data structures you know, for instance, a tuple to join a passenger and a seat, a dictionary for the occupation of the airplane, sets for the characters of the menu etc.

You will realize that the more modular is a program, the easier it is to implement.

Consider working in small groups, where each participant contributes one or more modules.