HANDLING DATA — PART 2

The objective of this sequence of works is to be able to handle data stored in different formats.

Last week, we have seen how to handle data through arrays, and then how to use dictionaries. For those of you who did not finish the work, please try to work on it 20 minutes, and then look at the elements of correction I uploaded on http://www.barsamian.am/2020-2021/S6ICT/TP10_Handling_data_1_correction.pdf.

This week, the goal is to learn how to handle data formatted into a specific way. Last week, the contact data was organized in a very simple fashion:

Line1: Name of contact 1

Line2: Phone of contact 1

Line3: Name of contact 2

Line 4: Phone of contact 2

etc.

This week, we will work on the ".csv" format. csv means comma-separated values. This means that on each line you have different values (as opposed to only one value per line, last week), and that those values are (usually) separated by commas (","). In fact, in the file we will work on, the values are separated by semi-colons (";").

The basic way to read the file is still the same than last week: we will open the file, and read it line by line. We will just see this week a more convenient way to loop through the lines, and we will see how to make Python understand that we have different values on each line.

In this problem, we want to handle a file containing GPS coordinates of French cities. The file is in French, but it should not be a problem. You can see the beginning of the file on Figure 1, and you can download it at http://www.barsamian.am/2020-2021/S6ICT/TP11_Cities.csv.

```
Code_commune_INSEE;Nom_commune;Code_postal;Libelle_acheminement;Ligne_5;coordonnees_gps
80355;FRESNEVILLE;80140;FRESNEVILLE;;49.9469630616, 1.753960976
80365;FRICAMPS;80290;FRICAMPS;;49.7720118421, 1.95186211928
80368;FRIVILLE ESCARBOTIN;80130;FRIVILLE ESCARBOTIN;;50.0912781795, 1.52364516053
80379;GLISY;80440;GLISY;;49.8341850031, 2.39954269272
80387;GRATTEPANCHE;80680;GRATTEPANCHE;;49.8142899245, 2.29952854065
80393;GRUNY;80700;GRUNY;;49.7015900422, 2.77539756139
80399;GUIGNEMICOURT;80540;GUIGNEMICOURT;;49.872686231, 2.06894684401
```

Figure 1: File containing GPS coordinates of French cities.

A skeleton of python code to manipulate this file is given in Listing 1, and can be downloaded from http://www.barsamian.am/2020-2021/S6ICT/TP11_Cities.py.

```
1  city = input("What city are your searching for? ")
2
3  # "iso-8859-1" is a common encoding. On Linux, the standard encoding is
4  # "utf-8" and on Windows you can also encounter "cp1252".
5  f = open("TP11_Cities.csv", "r", encoding="iso-8859-1")
6  # strip() removes blank characters at the beginning and the end of the string,
7  # here in particular the end of line characters left by readline()
8  for line in f:
9      values = line.strip().split(";")
10     # add your code here !
11 f.close()
```

Listing 1: Skeleton code to handle our cities file.

1. The `split` method transforms a string into an array of strings. Each string of the array is a sub-string of the original string, contained in two consecutive places where the pattern (here, ";") is found in the original string.

   For example, `"a+24x+30-53t+10".split("+")` is an array of size 4 (there are 3 "+" in the string, thus they separate the string in 4 distinct sub-strings), where the first cell is `"a"`, the second cell is `"24x"`, the third one is `"30-53t"` and the last one is `"10"`. In other words:

   `"a+24x+30-53t+10".split("+")` is the array `["a", "24x", "30-53t", "10"]`.

   When the python program reads the first line of the csv file, what is the content of the variable `values` after the instruction `values = line.strip().split(";")`?

2. You want to find the line containing information about a particular city (e.g., you're looking for LESBOEUFS). Modify the python file to print the value of the variable `values` when you have found the correct city. Which cell of the variable `values` has to be read if you want to extract its GPS coordinates?

3. Modify the program to extract the GPS coordinates of two cities given as input by the user. Then, have the program print the distance between the two cities.

   Hint: the cell that holds the GPS coordinates can itself be split, because it is always "latitude, longitude".

   Hint2: some explanation on how to deduce the distance from the coordinates can be found on the following link:
   https://stackoverflow.com/questions/365826/calculate-distance-between-2-gps-coordinates