

HANDLING DATA — PART 3

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

Last week, we have discovered the “.csv” format. Data encoded in this fashion can be seen as an array of arrays (or a 2d array, a matrix, or simply put, data that can be viewed as a tabular in a spreadsheet). Usually, each line in the file represents an entity (for example, a city), and each entity has different characteristics (for example: postal code, name...): one per column (two consecutive column are separated by commas (“,”) or, for French, by semi-colons (“;”)).

After the Carnaval holidays, we will learn a new language, “SQL”, to be able to express requests on this kind of data. This week, the goal is to implement some of the requests we will use, and to manipulate files.

First, we recall that we work with 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
```

Figure 1: File containing GPS coordinates of French cities.

The names of cities in this file are without accents, without special characters (no hyphen, no apostrophe...), and use capital letters only. A python code to manipulate this file, as it was needed last week, is given in Listing 1, and can be downloaded from http://www.barsamian.am/2020-2021/S6ICT/TP11_Cities_bis.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    name = values[1]
11    if (city.upper() == name):
12        print("The GPS coordinates of " + name + " are: " + values[5])
13 f.close()
```

Listing 1: Skeleton code to handle our cities file.

As two weeks ago, there is a first problem: different French cities have the same name! For example “Annay” is the name of two different cities.

1. If the name, alone, is not enough to uniquely distinguish French cities, what do you suggest to uniquely distinguish them?

To help the users of your program, we will add information about the “département” where the city lies. Each French département has an identifier, identifiers that are recalled in another file. The file is in French, but it should not be a problem. You can see the beginning of the file on Figure 2, and you can download it at http://www.barsamian.am/2020-2021/S6ICT/TP12_Departements.csv.

2. Modify the program to add the name of the département of the city, when you print it. It will help users know which “Annay” is the one they are looking for.

```
code_departement,nom_departement,code_region,nom_region
01,Ain,84,Auvergne-Rhône-Alpes
02,Aisne,32,Hauts-de-France
03,Allier,84,Auvergne-Rhône-Alpes
04,Alpes-de-Haute-Provence,93,Provence-Alpes-Côte d'Azur
05,Hautes-Alpes,93,Provence-Alpes-Côte d'Azur
```

Figure 2: File containing identifiers of French départements.

3. Given the name of a French “région”, list all the départements that are part of this région.
4. List all the cities in the cities file that have the same name of another French city.
5. List all the cities that are to the south of Valence. Create another .csv file that will receive exactly the same information as the original one, except that only those cities are kept.