11 Importing data from a csv file.

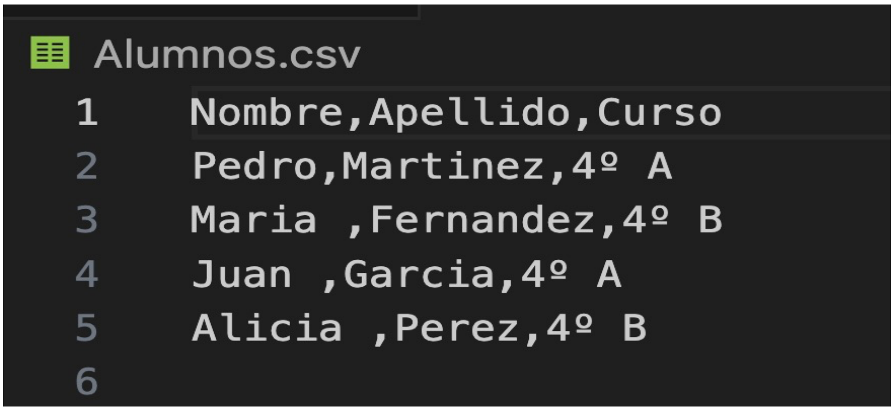
In previous chapters, we have explained how to read and write to text files using the different functions **open(), read(), readline(), readlines() write() and writelines().**

In this chapter we will see how to read and write csv files. First we will see the csv **library** which is the one that has implemented classes to read and write in csv file. We will use the following methods:

reader() -> To read data from the csv file. Creates the object to read each row of data in the csv file, returning each as a string list

writer() -> To write data to a csv file. Creates the object to write data, this object has the **writerow() method** that writes a row in the file.

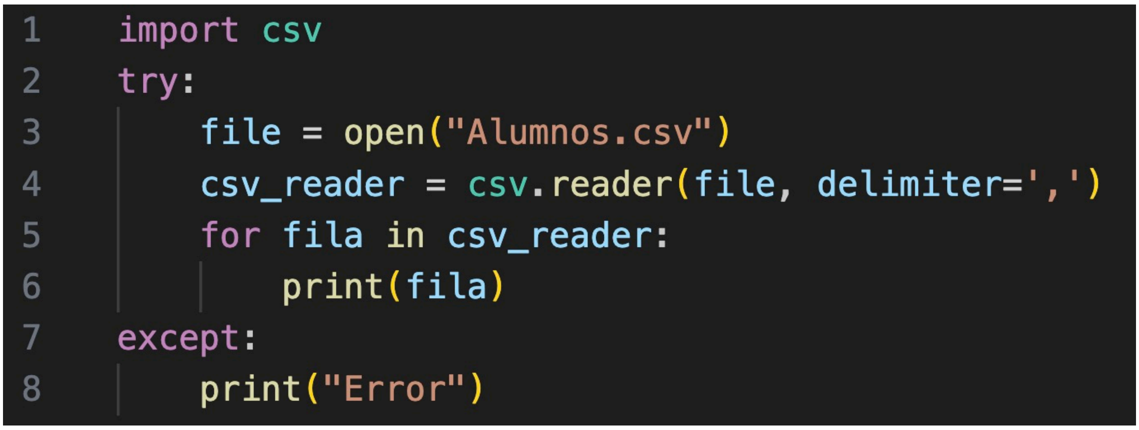
Let's look at the code and explain line by line that is being done. We have the following csv file.



Alumnos.csv

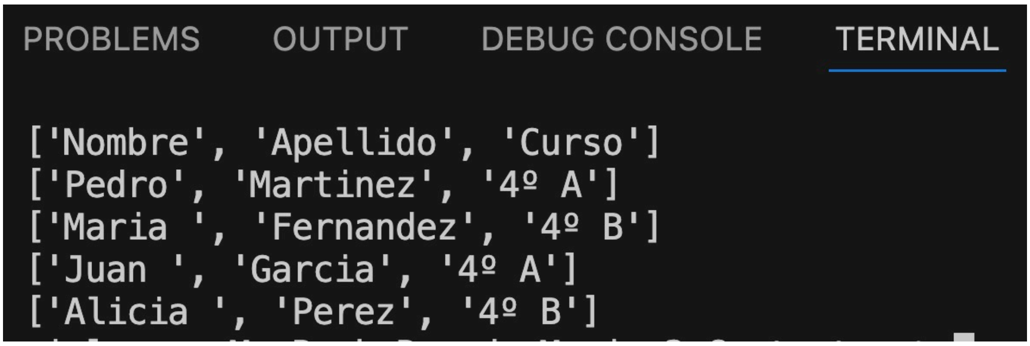
```
Nombre,Apellido,Curso
Pedro,Martinez,4º A
Maria ,Fernandez,4º B
Juan ,Garcia,4º A
Alicia ,Perez,4º B

```

The code we need to read this file is as follows:



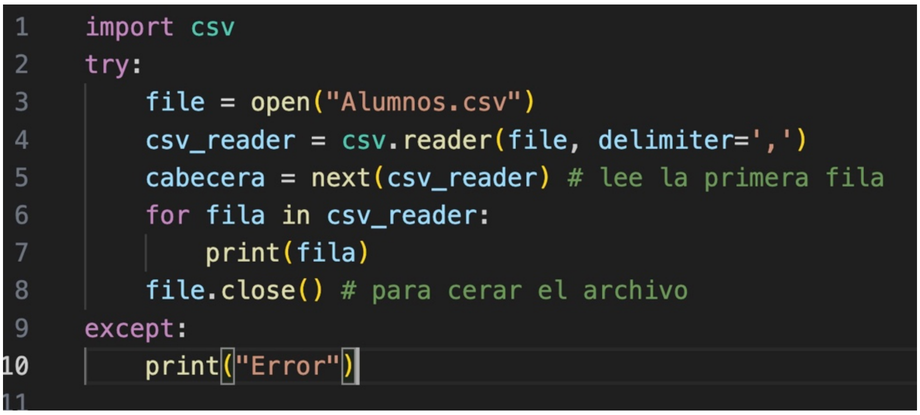
```
1  import csv
2  try:
3      file = open("Alumnos.csv")
4      csv_reader = csv.reader(file, delimiter=',')
5      for fila in csv_reader:
6          print(fila)
7  except:
8      print("Error")
```

The first thing we have to do is import the csv library**, which allows us to use the objects to work with the csv** files**. We do this in line 1. Then in the try** block **with the open() function we are opening the csv file in read mode, line 3. In line 4 we are reading the file with the csv.reader() function**, passing to it the parameters of the file that we have opened (file) and the type of field delimiter, in our case the fields are delimited by ','. As **csv.reader(), returns us a list for each file row we can iterate and show through the terminal what we have read. What would be shown in the terminal is as follows:**



```
['Nombre', 'Apellido', 'Curso']
['Pedro', 'Martinez', '4º A']
['Maria ', 'Fernandez', '4º B']
['Juan ', 'Garcia', '4º A']
['Alicia ', 'Perez', '4º B']
```

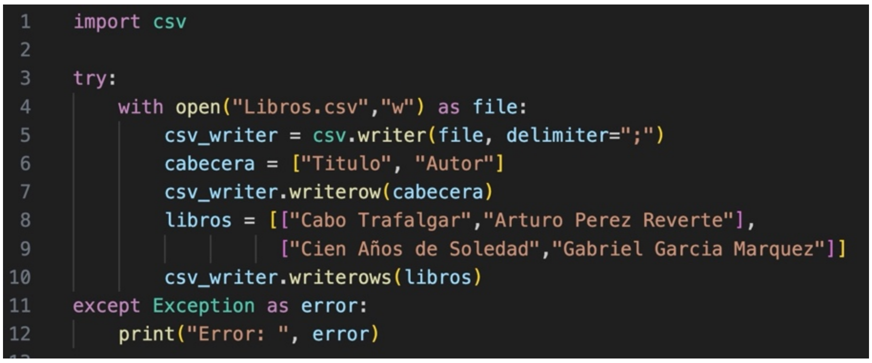
As we see, it shows us the contents of the file, as a list of text string for each of the rows of the file, being the first the name of the columns, if we did not want to show the names of the columns we would modify the code as follows:



```
1   import csv
2   try:
3       file = open("Alumnos.csv")
4       csv_reader = csv.reader(file, delimiter=',')
5       cabecera = next(csv_reader) # lee la primera fila
6       for fila in csv_reader:
7           print(fila)
8       file.close() # para cerar el archivo
9   except:
10      print("Error")
```

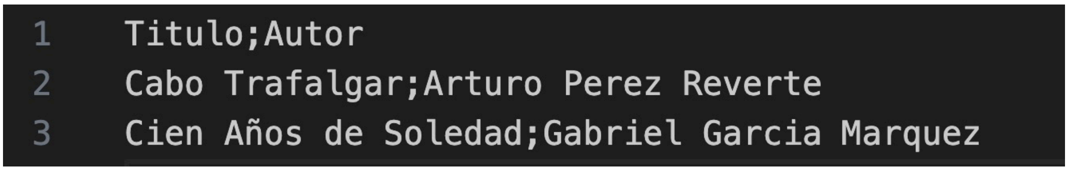
As you can see we have added line 5 the next(csv_reader) command**, this is what it does reads the first row, therefore when we run the for** command from line 6 will no longer show us in the terminal the first row. Once we have finished working with the open file we will have to use the function **file.close()** to close the file.

Now that we can read data from a csv **file**, let's add data to a csv file**.**



```
import csv

try:
    with open("Libros.csv","w") as file:
        csv_writer = csv.writer(file, delimiter=";")
        cabecera = ["Titulo", "Autor"]
        csv_writer.writerow(cabecera)
        libros = [["Cabo Trafalgar","Arturo Perez Reverte"],
                  ["Cien Años de Soledad","Gabriel Garcia Marquez"]]
        csv_writer.writerows(libros)
except Exception as error:
    print("Error: ", error)
```

In the code above, we have modified a little the way to open the file, we are using **with open() as fileen instead of file** = open(), **the explanation is that when we use** with open() as file**, we do not need to call the function** file.close(),**the file closes automatically, when we stop working with it, in our code after running line 10. We are also using both the writerow()** and **writerows(**) functions. The first adds only one row, while the second adds as many rows as we have defined lists, in our case two. The file generated by the code is as follows:



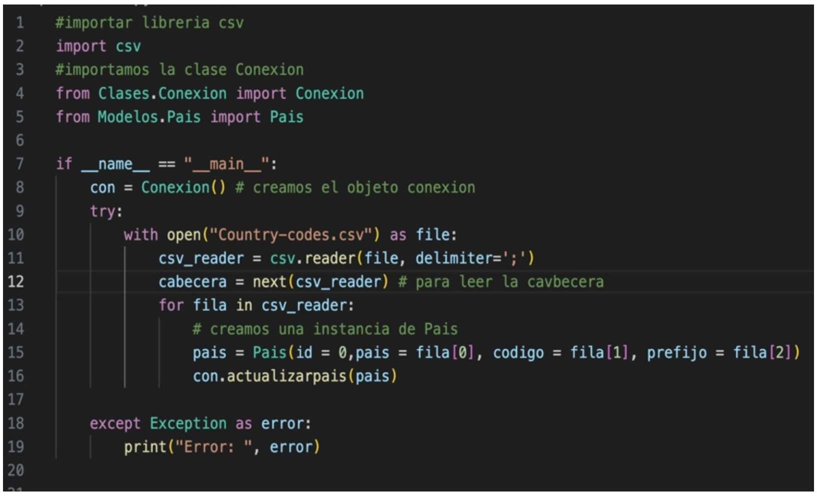
```
Titulo;Autor
Cabo Trafalgar;Arturo Perez Reverte
Cien Años de Soledad;Gabriel Garcia Marquez
```

The exercise that we are going to perform next is to read from a csv file of countries which has the following columns: country name, country code and telephone prefix and we will write to our table the table of countries of the database SQLite.

Also, in another chapter we have created the class **Connection,** where we have defined the different methods to be able to connect and perform the operations of adding, updating and deleting data in the tables, so let's use it.

We add a new file to which we will call **ImportarPaises.py**, this will be the file that we will run to import the countries into our table. The first thing we have to do is import the **csv** library in order to use the reader() object**. We also need to import the classes** Conexion **y** Pais**, which we have defined.**

The code we are going to create is as follows:



```
1   #importar libreria csv
2   import csv
3   #importamos la clase Conexion
4   from Clases.Conexion import Conexion
5   from Modelos.Pais import Pais
6
7   if __name__ == "__main__":
8       con = Conexion() # creamos el objeto conexion
9       try:
10          with open("Country-codes.csv") as file:
11              csv_reader = csv.reader(file, delimiter=';')
12              cabecera = next(csv_reader) # para leer la cavbecera
13              for fila in csv_reader:
14                  # creamos una instancia de Pais
15                  pais = Pais(id = 0,pais = fila[0], codigo = fila[1], prefijo = fila[2])
16                  con.actualizarpais(pais)
17
18      except Exception as error:
19          print("Error: ", error)
20
```

The imports that we need do it with the code of lines 2, 4 and 6. In the block of execution on line 8, we create an object of the class Connection to connect to the database and be able to access its functions. Now in the try **block** , the first thing is to open the csv **file** that we are going to process, the parameter that we pass to the open() **function** is the name of the file, we do not pass the mode that we want to open it, because by default it is read. In line 11 we call the function **csv.reader()** to read the data of the file. In the next line, we **read the first row of the file, in the case of the** Country-codes.csv file, is the header information, which we do not want to save in the table. In **line 13 with the command** we will use the data that we have read and for **each row we create an instance of the Pais**con class with its values (line 15) and call the method actualizarpais() **that we have defined in the Conexion** class **passing as parameter the country data** (line 16).

With this code we have imported the data from the csv**y file** we have saved it in the table of landscapes in the SQLite database.