CSV

March 30, 2022

Ejecuta esta línea para cargar los archivos necesarios para el notebook

[1]:
```
!wget https://raw.githubusercontent.com/cosmolejo/dataRepo/master/
↪starwars_characters.csv
```

```
--2022-03-30 03:52:28--  https://raw.githubusercontent.com/cosmolejo/dataRepo/ma
ster/starwars_characters.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.108.133, 185.199.111.133, 185.199.109.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 5462 (5.3K) [text/plain]
Saving to: ‘starwars_characters.csv’

starwars_characters 100%[===================>]   5.33K  --.-KB/s    in 0s

2022-03-30 03:52:30 (38.1 MB/s) - ‘starwars_characters.csv’ saved [5462/5462]
```

1 Manejo de archivos 3

Veamos el segundo tipo de archivo que nos falta en esta sección, los archivos ***CSV***

1.1 Archivos CSV

Un archivo CSV (valores separados por comas) es un archivo de texto que tiene un formato específico que permite guardar los datos en un formato de tabla estructurada.

- Como su nombre lo dice, cada dato está separado por comas.
- Por convención, se usa la primera línea como cabecera del archivo, es decir, en la primera línea se guarda el nombre de cada una de las columnas para poder identificarlas.
- Cada línea es un registro independiente, es decir, que representa un objeto único de datos implícitamente estructurados en una tabla.
- Si en algún registro falta algún dato, se debe dejar el campo vacío entre las dos comas que lo rodearían.

```
nombre,edad,sexo
'Hugo',35,'M'
```

```
'Luisa',,'F'
'Paco',24,'M'
```

1.2 Lectura de archivos

Si bien este tipo de archivos se puede leer sin problemas con las funciones nativas de Python, usando la función `split(',')` para que separen cada línea por las comas. Python también cuenta con una librería que simplifica el proceso de lectura y escritura.

Empecemos por importar dicha librería.

[2]:
```
import csv
```

Luego, abrimos el archivo normalmente con la sentencia `with` y en su interior, leeremos el archivo con la función `reader` de la librería.

[3]:
```
cabecera = []
registros = []
with open('starwars_characters.csv', 'r') as csvfile:
    lector = csv.reader(csvfile)

    #para sacar la cabecera usamos la función next
    cabecera = next(lector)

    # extraemos cada registro con un ciclo for
    for fila in lector:
        registros.append(fila)

    print('total de registros encontrados', lector.line_num)

print('cabecera: ', cabecera)
print('primeros 5 registros')
for i in range(5):
    print(registros[i])
```

```
total de registros encontrados 88
cabecera:  ['name', 'height', 'mass', 'hair_color', 'skin_color', 'eye_color',
'birth_year', 'gender', 'homeworld', 'species']
primeros 5 registros
['Luke Skywalker', '172', '77', 'blond', 'fair', 'blue', '19BBY', 'male',
'Tatooine', 'Human']
['C-3PO', '167', '75', 'NA', 'gold', 'yellow', '112BBY', 'NA', 'Tatooine',
'Droid']
['R2-D2', '96', '32', 'NA', 'white, blue', 'red', '33BBY', 'NA', 'Naboo',
'Droid']
['Darth Vader', '202', '136', 'none', 'white', 'yellow', '41.9BBY', 'male',
'Tatooine', 'Human']
['Leia Organa', '150', '49', 'brown', 'light', 'brown', '19BBY', 'female',
'Alderaan', 'Human']
```

La función `next` nos permite solicitarle a objetos especiales de Python como `range` o el objeto creado por `csv.reader`, llamados iterables que nos devuelvan la fila actual y avanzar el iterable a la siguiente fila, descartando (por decirlo de algún modo) la fila que nos entregó. Como la primera fila de nuestro archivo csv contiene las cabeceras (o nombres de campos) las guardamos en una lista llamada `cabecera`. Estos iterables son un tema de programación más avanzado, por lo cual no entraremos demasiado en detalles al respecto, ya que se sale del alcance de nuestro curso.

Luego de usar `next` la lista de filas almacenada en `lector` ya no tiene la cabecera, por lo que podemos usar un simple ciclo `for` para guardar en una lista cada registro por separado. Como vemos en el ejemplo, el resultado de este proceso es una lista de listas, por lo que acceder a cualquiera de los datos es tan simple como usar la notación de doble corchetes `[ ][ ]`

El iterable `lector` almacena el número de registros que leyó durante el ciclo en la variable `line_num` a la cual accedemos con `lector.line_num` siempre que la necesitemos.

1.3 Escritura a un archivo CSV

Para escribir un archivo en csv, creamos un objeto con la función `writer` de la librería csv, el cual se encargará de almacenar los registros que le entreguemos con `writerow` o `writerows` de forma ordenada.

> `writerow` nos permite escribir un solo registro a la vez, mientras que `writerows` nos permite entregarle una lista con todos los registros, acelerando el proceso de escritura.

[ ]:
```
cabecera = ['Nombre', 'Apellido', 'Carrera', 'Semestre','Promedio']

datos = [['Paco','Peralta','Psicología',8,4.0],
         ['Gustavo','Lopez','Física',5,3.8],
         ['Rosario','Márquez','Medicina',9,4.7],
         ['Roberto','Florez','Ing. Sistemas',7,3.5]]

with open('notas_universidad.csv','w') as csvfile:
    escritor = csv.writer(csvfile)

    #escritura de la cabecera
    escritor.writerow(cabecera)

    #escritura de registros
    escritor.writerows(datos)
```

Verifiquemos el archivo:

[ ]:
```
with open('notas_universidad.csv','r') as salida:
    print(salida.read())
```

```
Nombre,Apellido,Carrera,Semestre,Promedio
Paco,Peralta,Psicología,8,4.0
Gustavo,Lopez,Física,5,3.8
Rosario,Márquez,Medicina,9,4.7
```

```
Roberto,Florez,Ing. Sistemas,7,3.5
```