

# Clase5

April 13, 2021

## 1 Seminario de Lenguajes - Python

### 1.1 Cursada 2021

#### 1.1.1 Clase 5

### 1.2 Comentamos el ejemplo que dejé en el video: ¿algo está mal?

```
[ ]: def leo_caracteres():  
    f = open("imagine.txt", "r")  
    for x in f.read():  
        print(x)  
    f.close()
```

## 2 CSV: ¿más formatos?

- CSV (Comma Separated Values).
- Es un formato muy común para importar/exportar desde/hacia hojas de cálculo y bases de datos.
- Ejemplo:

nombre,ciudad,ref

William Campbell,La Plata,[www.instagram.com/williamcampbellok](http://www.instagram.com/williamcampbellok)

Buendia,La Plata,<https://buendia.bandcamp.com/>

Lúmine,La Plata,<https://www.instagram.com/luminelp/>

- +Info: <https://docs.python.org/3/library/csv.html>
- [PEP 305](#)

## 3 Datasets

- Hay muchos datasets disponibles de muchas temáticas:
- En nuestro país:
  - Datos de [Argentina](#)
  - Datos de [CABA](#)
  - Datos de [La Plata](#)
- Otros:
  - <https://data.world/>
  - <https://www.kaggle.com/>

- <https://www.imdb.com/interfaces/>
- Y muchos más...

## Muchos son datos abiertos, pero otros... no tanto...

### ¡PRESTAR ATENCIÓN a la licencias y requisitos para su uso!

## 4 ¿Qué vemos en netflix?

Vamos a trabajar con el archivo: [netflix\\_titles.csv](#)

```
[ ]: import csv

archivo = open("ejemplos/clase5/netflix_titles.csv", "r")
csvreader = csv.reader(archivo, delimiter=',')

#encabezado = csvreader.__next__()
encabezado = next(csvreader)

print(encabezado)
print("Titulo"+" "*35+"Director")
for linea in csvreader:
    if linea[1] == "TV Show" and linea[5] == "Argentina":
        print(f"{linea[2]:<40} {linea[3]}")

archivo.close()
```

## 5 El módulo cvs

- Hay que importarlo.
- **csv.reader**: cea un objeto “iterador” que nos permite recorrer las líneas del archivo.
- ¿De qué tipo es línea?
- ¿Por qué incluimos el parámetro **delimiter**? ¿Dialectos?

## 6 Otra solución ...

```
[ ]: import csv

archivo = open("ejemplos/clase5/netflix_titles.csv", "r")
csvreader = csv.reader(archivo, delimiter=',')

shows_ar = filter(lambda x: x[5] == "Argentina" and x[1] == "TV Show",
                  ↪ csvreader)
for elem in shows_ar:
    print(f"{elem[2]:<40} {elem[3]}")

#print(shows_ar)
archivo.close()
```

## 7 Creamos nuestro archivo csv de bandas de música

- **csv.writer:** retorna un objeto que convierte los datos con los que trabajamos en el programa en cadenas con el formato delimitadas con el separador correspondiente.

```
[ ]: import csv
import json

archivo = open("bandas.txt")
archivo_csv = open("bandas.csv", "w")

bandas = json.load(archivo)
datos_a_mostrar = json.dumps(bandas, indent=4)
print(datos_a_mostrar)

writer = csv.writer(archivo_csv)
writer.writerow(["Nombre", "Ciudad de procedencia", "Referencias"])
for banda in bandas:
    writer.writerow([banda["nombre"], banda["ciudad"], banda["ref"]])

archivo.close()
archivo_csv.close()
```

## 8 Lo leemos

```
[ ]: archivo_csv = open("bandas.csv", "r")
csvreader = csv.reader(archivo_csv, delimiter=',')

for linea in csvreader:
    print(linea)

archivo_csv.close()
```

## 9 Primer desafío

- Dado el conjunto de datos con series y películas de Netflix, queremos:
  - 1- guardar en otro archivo las películas agregadas en el año 2020.
  - 2- los cinco (5) países con más producciones en Netflix.

```
[ ]: # Solución.....
```

### 9.1 Los que quieran, compartir solución con @clauBanchoff

- Recuerden mandarme mensaje para revisar.

## 10 ¿Saben qué son las GUI?

### 10.1 GUI: Graphical User Interface

## 11 GUI en Python

- Vamos a usar [PySimpleGUI](#)
- Es un **framework** bastante simple para desarrollar interfaces gráficas en Python.
- Es software libre: <https://github.com/PySimpleGUI/PySimpleGUI>
- Se instala con pip: **pip install pysimplegui**
- Vamos a ir de a poco...
- Más info en: <https://pysimplegui.readthedocs.io/en/latest/cookbook/>
- También en <https://realpython.com/pysimplegui-python/>

## 12 ¿Con qué elementos podemos trabajar?

## 13 Popups: las ventanas más sencillas

```
[ ]: import PySimpleGUI as sg
sg.Popup('Mi primera ventanita')
```

- **import PySimpleGUI as sg**, permite acceder a los recursos por el nombre **sg**.

```
[ ]: import PySimpleGUI as sg

sg.Popup('Mi primera ventanita', button_color=('black', 'red'))

sg.PopupYesNo('Mi primera ventanita', button_color=('black', 'green'))

sg.PopupOKCancel('Mi primera ventanita', button_color=('black', 'grey'))

texto = sg.PopupGetText('Titulo', 'Ingresá algo')
sg.Popup('Resultados', 'Ingresaste el siguiente texto: ', texto)
```

## 14 Creamos una ventana en PySimpleGUI

```
[ ]: import PySimpleGUI as sg

sg.Window(title="Hola Mundo!", layout=[[]], margins=(100, 50)).read()
```

- **margins**: tamaño de la ventana en pixeles.
- **read()**: devuelve los eventos que se produzca al activar la ventana.

## 15 Sobre los eventos

```
[ ]: import PySimpleGUI as sg

layout = [[sg.Text("Hola Mundo!"), [sg.Button("OK")]]

window = sg.Window("Primer Demo", layout, margins=(200, 150))

while True:
    event, values = window.read()

    if event == "OK" or event == sg.WIN_CLOSED:
        break

window.close()
```

## 16 ¿Cómo recuperamos los valores ingresados desde la UI?

```
[ ]: import PySimpleGUI as sg

layout = [ [sg.Text('Ingresá primer valor'), sg.InputText()],
           [sg.Text('Ingresá segundo valor'), sg.InputText()],
           [sg.Button('Ok'), sg.Button('Cancel')] ]

window = sg.Window("Segundo Demo", layout, margins=(200, 150))

while True:
    event, values = window.read()

    if event == "Cancel" or event == sg.WIN_CLOSED:
        break
    print('Datos ingresados: ', values)

window.close()
```

## 17 Layout: ¿cómo organizamos la UI?

Representa al esquema o diseño de nuestra UI: **cómo se distribuyen los elementos en la UI.**

```
[ ]: layout = [ [sg.Text('Ingresá primer valor'), sg.InputText()],
               [sg.Text('Ingresá segundo valor'), sg.InputText()],
               [sg.Button('Ok'), sg.Button('Cancel')] ]
```

- ¿De qué tipo es la variable layout?
- ¿Qué elementos estamos incluyendo?

## 18 Elementos de la UI

- Acá van algunos disponibles en PySimpleGUI
  - Buttons: File Browse, Folder Browse, Color chooser, Date picker, etc.
  - Checkbox, Radio Button, Listbox
  - Slider, Progress Bar
  - Multi-line Text Input, Scroll-able Output
  - Image, Menu, Frame, Column, Graph, Table
- Referencias: <https://pysimplegui.readthedocs.io/en/latest/call%20reference/#element-and-function-call-reference>

## 19 Agreguemos elementos

```
[ ]: import PySimpleGUI as sg

layout = [[sg.Input('Ingresa algo')],
          [sg.Listbox(values=('Item 1', 'Item 2', 'Item 3'))],
          [sg.OK()]]

window = sg.Window("Elementos básicos", layout, margins=(200, 150))
event, values = window.read()

sg.Popup(event, values, line_width=200)
```

```
[ ]: sg.ChangeLookAndFeel('DarkAmber')

layout = [[sg.Listbox(values=('Item 1', 'Item 2', 'Item 3'),
    ↪background_color='yellow', size=(20,3)),
          [sg.Input('Last input')],
          [sg.ColorChooserButton(" Elegi color")],
          [sg.OK()]] ]
```

## 20 PySimpleGUI en GitHub

- En [PySimpleGUI.py](#) hay mucha información y referencias útiles.
- Por ejemplo, hay [varios ejemplos](#) que pueden servir de base para nuestros desarrollos:

## 21 ¡¡Las ventajas del software libre!!

## 22 Hablemos del trabajo integrador