

Práctica 3 - Seminario de Lenguaje Python 2022

April 20, 2022

Objetivos:

- Manejo de archivos
- Módulo os
- Módulo datetime

1 Archivos CSV

- Datos de Netflix

Veamos el archivo `netflix_titles.csv` con el que estuvieron trabajando en la teoría

```
[ ]: path_files = "files"
      archivo_net = "netflix_titles.csv"
```

- ¿Cómo accederían al archivo?
- ¿En qué directorio guardarían estos datos?
- ¿Qué cosas hay que tener en cuenta?

```
[ ]: import csv
```

```
[ ]: import os.path
      import os
      path_arch = os.path.join(os.getcwd(), path_files)
```

- Vemos los datos que tenemos, consultando los nombres de las columnas

```
[ ]: archivo = open(os.path.join(path_arch, archivo_net), "r")
      data_net = csv.reader(archivo, delimiter=',')
      header , datos = next(data_net), list(data_net )
      header
```

- Recorremos el archivos para conocer las información que contiene

```
[ ]: for linea in datos:
      print(linea)
```

1.1 Ejercicios

1. Encontrar qué tipo de shows tiene un país determinado.

- Realizar una función que informe todos los países que existen.
- Realizar una función que dado un país informe si es parte de la línea del show pasado como argumento. Nota: utilice las funciones vistas de lambda(utilizando la función definida), map para informar los tipos de shows (valores únicos) en que participa un país.

Analizar:

- ¿En qué número de columna está el país?
 - Como en algunos casos hay varios países en un show debemos separarlos y quedarnos con valores únicos.
2. Informe la lista de países del archivo en orden alfabéticamente creciente.
 3. Informe los shows de un año determinado, realice una función que reciba un año y la línea como argumentos.

2 Fechas - datetime

Documentación [datetime](#)

```
[ ]: import datetime
```

El módulo datetime crea un objeto con el cual podemos realizar operaciones para cálculo de fechas.

```
[ ]: x = datetime.datetime.now()
x
```

- Como saber qué número de día de la semana es hoy

```
[ ]: nro_dia = datetime.datetime.today().weekday()
nro_dia
```

- Para obtener los días de la semana en castellano podemos utilizar una estructura auxiliar:

```
[ ]: dias_semana = ['lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',
↳ 'domingo']
dias_semana[nro_dia]
```

- Consultar algún dato en particular, como la hora

```
[ ]: x.hour
```

- Para guardar los datos en archivos se debe guardar en string indicando el formato en que queremos guardarla.

```
[ ]: horario_juego = datetime.datetime.now().strftime("%d/%m/%Y,%H:%M:%S")
horario_juego
```

2.1 Ejercicios:

- En base al [archivo](#) analizar las fechas de los logs:

```
[ ]: import os
```

```
[ ]: logs = 'BBB_nuevo.csv'
with open(os.path.join(path_arch, logs)) as logs_moodle:
    data_logs = csv.reader(logs_moodle, delimiter=',')
    header , logs_recurso = next(data_logs), list(data_logs )
for linea in logs_recurso:
    print(linea[0])
```

Analicemos cómo calcular el día de la semana que corresponde el log, el dato está guardado como string y para hacer consults debemos convertirlo a objeto *datetime*. El segundo argumento de la función *strptime* corresponde al *formato*, donde debemos indicar cómo se encuentra cada dato:

- %d = día
- %m = mes
- %Y = año, si fuera la forma corta(22) corresponde %y
- luego tiene un espacio
- %H = hora
- %M = minutos

```
[ ]: formato = "%d/%m/%Y %H:%M"
print(datetime.datetime.strptime(linea[0], formato).weekday())
```

```
[ ]: dias_semana = ['lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',
↪ 'domingo']
nro_dia = datetime.datetime.strptime(linea[0], formato).weekday()
dias_semana[nro_dia]
```

```
[ ]: datetime.datetime.strptime(linea[0], formato).weekday()
```

4. Indique los días de la semana que más registros hubo:
5. Calcule cuántos días pasaron entre el primer registro y el último.