¿Como Pasar de un Archivo en Formato CSV a un Diccionario Funcional?

Para este ejercicio emplearemos un enfoque mucho más orientado a la programación convencional. Este método se emplea en determinados casos cuando no deseemos gastar tantos recursos con librerías externas.

PROCEDIMIENTO

**PASO 01: “Abrir dicho archivo”**

**-**Por supuesto antes de comenzar debemos abrir nuestro respectivo archivo con la función **(open),** preferiblemente asignarlo a una variable.

-OJO: Debemos recordar que es mejor trabajar un csv de forma local.

**PASO 02: “Comprobar iterando el archivo”**

-Dicho Archivo es un objeto iterable, el cual podremos recorrer. Cada elemento que nos devuelva sería pues la propia línea de strings.

-De forma que en sí podemos comprobarlo, lo ideal sería ver las primeras 3 líneas.

**PASO 03: “Generamos Nuestro Diccionario”**

-Sencillamente creamos un diccionario, con los mismos apartados (renombrados) que contenga nuestro presunto archivo CSV

**PASO 04: “Empezaremos dándole formato a las respectivas líneas del archivo”**

**-**Empezamos saltando la primera línea, no nos interesa debido a que el diccionario, ya las ha renombrado.

-Volvemos a iterar las “líneas” en el archivo.

-Haremos que cada línea sea igual a sí misma, pero sin los elementos extremos, esto en teoría funciona para eliminar el salto de línea. **FUNCION STRIP ()**

**-**Ahora identificamos el separador, al obtenerlo crearemos una lista, la cual tendrá como elementos a aquellos separados. SPLIT(“separador”)

-Teniendo la lista completa, procedemos a rellenar al diccionario.

-Para ello usamos índice, elemento enumerate.

-Donde cada lista de diccionario por su llave, le agregamos su campo que corresponde con su índice.

-Terminado esto cerramos esto con archivo. close ()

**RECOMENDACIÓN: Podemos Usar Pandas para “Comprobar” Nuestros resultados y no olvidemos ejecutar una sola vez**

**Ejemplo de Código:**

import pandas as pd

tabla = pd.DataFrame(dicc\_emisiones)

tabla.iloc[35]["anio"]

**PASO 05: “CASTEAREMOS LOS DATOS SEGÚN NOS CONVENGA”**

-Con pandas por ejemplo comprobamos que algunos datos numéricos, presentan problemas:

* El año es un strings
* El co2 tiene un (.) que separa los millares y (,) que separa los decimales - str
* El co2\_percapita usa (,) para separar los decimales, y es un string

-Para castear nuestros datos por columna procedemos a:

--Recorrer un índice hasta el rango del largo de nuestra lista/columna diccionario

-Procedemos a castear cada elemento de la siguiente forma:

Dicc[“llave”][i] = int (Dicc[“llave”][i])

**Paso 06: “Casteo Complejo, Dar Formato a los Datos para Operar con Ellos”**

**-**Este caso resulta muy similar a lo anterior, sin embargo, debemos jugar mucho más con los Strings

-Como en este caso operaremos con cada elemento, haremos un ennumerate, pero en la lista/columna que deseamos cambiar.

**RECOMENDACIÓN: Podemos Usar Pandas para “Comprobar” Nuestros resultados, por ejemplo, con sample, darnos cuenta sobre la naturaleza de la tabla**

**-**Tras algunas pruebas nos damos cuenta que co2 y co2\_percapita cuenta con gran cantidad de datos faltantes además de su falta de formato.

-Tomamos el elemento del ennumerate y lo modificaremos a nuestro gusto con tal de obtener un resultado válido.

-No olvidar por favor los valores faltantes.

**-ATENCION: Se debe tener cuidado a la hora de ejecutar estas líneas de código, ya que posiblemente se cambie el tipo de dato y a continuación volver a correrlo sería un grave error.**

**Paso 07: “Operaciones con Filtros”**

-Para operar según algunos parámetros, podemos con un índice recorrer nuestra lista o columna principal (La cual someteremos a nuestro análisis)

-Creamos nuestras condiciones, teniendo en cuenta los elementos solitarios, a través del índice por supuesto.

-Una vez con las condiciones, si TODAS se cumplen entonces usamos ese índice para operar sobre los elementos de la columna principal.

-De esta forma obteniendo la suma o lo que deseamos.