

Desafio Stark

Industrias Stark es principalmente una empresa de defensa que desarrolla y fabrica armas avanzadas y tecnologías militares.



Recientemente ha decidido ampliar su departamento de IT y para acceder a las entrevistas es necesario completar el siguiente desafío, el cual estará dividido en etapas. Cada semana recibiremos un nuevo pedido de parte de la empresa.



La empresa compartió con todos los participantes cierta información confidencial de un grupo de superhéroes. Y semana a semana enviará una lista con los nuevos requerimientos. Quien supere todas las etapas accederá a una entrevista con el director para de la compañía.

Set de datos

La información a ser analizada se encuentra en el archivo `data_stark.json`.

Desafío #05: (crear biblioteca stark_desafio_5.py)

Realizar las siguientes funciones

1. Primera Parte: Archivos

- 1.1. Crear la función 'leer_archivo' la cual recibirá por parámetro un string que indicará el nombre y extensión del archivo a leer (Ejemplo: archivo.json). Dicho archivo se abrirá en modo lectura únicamente y retornara un string con la información del mismo.

ATENCIÓN:Controlar las excepciones posibles en caso de presentarse alguna imprimir el mensaje de la misma y retornar **False**

Usar de Encoding UTF-8

- 1.2. Crear la función 'guardar_archivo' la cual recibirá por parámetro un string que indicará el nombre con el cual se guardará el archivo junto con su extensión (ejemplo: 'archivo.csv') y como segundo parámetro tendrá un string el cual será el contenido a guardar en dicho archivo. Debe abrirse el archivo en modo escritura+, ya que en caso de no existir, lo creara y en caso de existir, lo va a sobrescribir La función debera retornar **True** si no hubo errores, caso contrario **False** (**VALIDAR CON EXCEPCIONES**), además en caso de éxito, deberá imprimir un mensaje respetando el formato:
Se creó el archivo: nombre_archivo.csv

ATENCIÓN:Controlar las excepciones posibles en caso de presentarse alguna retornar false e imprimir un mensaje que diga:: 'Error al crear el archivo: nombre_archivo'

Donde nombre_archivo será el nombre que recibirá el archivo a ser creado, conjuntamente con su extensión.

Usar de Encoding UTF-8

1.3. Crear la función **generar_csv()**

La función va a recibir el nombre y extensión del archivo csv de los superhéroes (Puede ser ruta absoluta o relativa) y la lista de los mismos.

Si la lista no está vacía la función deberá guardar en un string la información en formato csv (separado con comas) con la cabecera correspondiente.

Una vez generado el string debería usar la función de 1.2 para guardar ese string generado al archivo.

Si la lista está vacía retornar **False**

Debería quedar algo como esto:

```
heroes.csv
1 nombre,identidad,empresa,altura,peso,genero,color_ojos,color_pelo,fuerza,inteligencia
2 Howard the Duck,Howard (Last name unrevealed),Marvel Comics,79.34999999999999,18.44999999999999,M,Brown,Yellow,2,
3 Rocket Raccoon,Rocket Raccoon,Marvel Comics,122.77,25.73,NB,Brown,Brown,5,average
4 Wolverine,Logan,Marvel Comics,160.69999999999999,135.21000000000001,M,Blue,Black,35,good
5 Black Widow,Natasha Romanoff,Marvel Comics,170.78999999999999,59.34000000000003,F,Green,Auburn,15,good
6 Mystique,Raven Darkholme,Marvel Comics,178.65000000000001,54.96000000000001,F,Yellow (without irises),Red / Orange,15,good
7 Spider-Man,Peter Parker,Marvel Comics,178.28,74.25,M,Hazel,Brown,55,high
8 Storm,Ororo Munroe,Marvel Comics,180.72,57.5,F,Blue,White,10,good
9 Gamora,Gamora Zen Whoberi Ben Titan,Marvel Comics,183.65000000000001,77.76999999999999,NB,Yellow,Black,85,good
10 Thing,Ben Grimm,Marvel Comics,183.55000000000001,225.41999999999999,M,Blue,No Hair,85,good
11 Thor,Thor Odinson,Marvel Comics,198.34999999999999,288.61000000000001,M,Blue,Blond,100,good
12 Colossus,Piotr Nikolaievitch Rasputin,Marvel Comics,226.13,225.52000000000001,NB,Silver,Black,85,good
13 Hulk,Bruce Banner,Marvel Comics,244.40000000000001,630.89999999999998,M,Green,Green,100,high
14 Groot,Groot,Marvel Comics,701.12,4.820000000000003,M,Yellow,,85,good
15 Daredevil,Matt Murdock,Marvel Comics,183.68000000000001,90.45000000000003,M,Blue,Red,15,good
16 Nick Fury,Nicholas Joseph Fury,Marvel Comics,185.88,99.28000000000001,M,Brown,Brown / White,15,good
17 Punisher,Frank Castle,Marvel Comics,183.25999999999999,90.90000000000006,M,Blue,Black,20,good
18 Star-Lord,Peter Jason Quill,Marvel Comics,188.25,79.20999999999999,M,Blue,Blond,20,good
19 Deadpool,Wade Wilson,Marvel Comics,188.0,95.31999999999999,M,Brown,No Hair,35,good
20 Captain America,Steve Rogers,Marvel Comics,188.00999999999999,108.94,M,blue,blond,20,good
```

1.4. Crear la función **leer_csv()** que va a recibir el nombre y extensión de donde se ubica el archivo a leer (Ruta absoluta o relativa)

La función se tiene que encargar de generar una lista de superhéroes en base al contenido de ese archivo csv que se le paso. Pueden usar la cabecera de ese csv para generar las claves de cada uno de los diccionarios.

La función debe retornar la lista de diccionarios si es que existe el archivo y sino False.

1.5. Crear la función **generar_json()** que va a recibir el nombre y extensión de donde se ubica el archivo a guardar (Ruta absoluta o relativa) , la lista de los superhéroes y el nombre de la lista.

Si la lista no está vacía debería guardar en un diccionario de una sola clave la lista de superhéroes,el nombre de clave debería ser la del tercer parámetro que sería el nombre de la lista.

EJEMPLO DE LLAMADA

```
generar_json("heroes.json", lista_personajes, "heroes")
```

COMO DEBERIA QUEDAR EL ARCHIVO

```
() heroes.json > ...
1  {}
2  "heroes": [
3    {
4      "nombre": "Howard the Duck",
5      "identidad": "Howard (Last name unrevealed)",
6      "empresa": "Marvel Comics",
7      "altura": 79.35,
8      "peso": 18.45,
9      "genero": "M",
10     "color_ojos": "Brown",
11     "color_pelo": "Yellow",
12     "fuerza": 2,
13     "inteligencia": ""
14   },
15   {
16     "nombre": "Rocket Raccoon",
17     "identidad": "Rocket Raccoon",
18     "empresa": "Marvel Comics",
19     "altura": 122.77,
20     "peso": 25.73,
21     "genero": "NB",
22     "color_ojos": "Brown",
23     "color_pelo": "Brown",
24     "fuerza": 5,
25     "inteligencia": "average"
26   },
27   {
28     "nombre": "Wolverine",
29     "identidad": "Logan",
30     "empresa": "Marvel Comics",
31     "altura": 160.7,
32     "peso": 135.21,
33     "genero": "M",
34     "color_ojos": "Blue",
35     "color_pelo": "Black",
36     "fuerza": 35,
37     "inteligencia": "good"
38   },
39   {
```

ATENCIÓN: Usar indent=4 al generar el Json

También tener en cuenta que los datos deben estar normalizados antes de generar el json.

- 1.6. Crear la función `leer_json()` que va a recibir el nombre y extensión de donde se ubica el archivo a leer (Ruta absoluta o relativa), y también el nombre de la lista a leer por ejemplo en la imagen anterior la lista está en la clave "heroes"
- Si el archivo existe debería leer el archivo json y retornar la lista obtenida.
- Si el archivo no existe debería retornar **False (USAR EXCEPCIONES)**

EJEMPLO DE LLAMADA

```
lista_personajes = leer_json("heroes.json","heroes.heroes.json")
```

2. Segunda Parte

- 2.1. Crear una función para ordenar héroes por alguna de las claves numéricas (altura, peso y fuerza) de manera ascendente
- 2.2. Crear una función para ordenar héroes por alguna de las claves numéricas (altura, peso y fuerza) de manera descendente.
- 2.3. Crear una función para ordenar héroes por alguna de las claves numéricas (altura, peso y fuerza). Preguntar al usuario si lo quiere ordenar de manera ascendente ('asc') o descendente ('desc') (**reutilizar funciones anteriores dependiendo del caso**)
3. Desarrollar los siguientes puntos (usando el `data_stark.py`) -> Se deben desarrollar funciones extra dependiendo de la situación o reutilizar funciones anteriores como normalizar o listar.
- 1-Normalizar datos (No debe dejar de entrar a las otras opciones)
 - 2-Generar CSV (Guardar la lista generada en otra variable)
 - 3-Listar heroes del archivo CSV ordenados por altura ASC (Validar si el mismo existe)
 - 4-Generar JSON (Guardar la lista generada en otra variable)

- 5-Listar heroes del archivo JSON ordenados por peso DESC (Validar si el mismo existe)
- 6-Ordenar Lista por fuerza (Se le tiene que preguntar al usuario si ordenar de manera ASC o DESC)
- 7-Salir