r = requests.get(url='https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json')

json\_readed = r.json()# Paso el archivo 'r' que estaba en la url usando el metodo json de la libreria requests. El metodo json le va a dar una estructura de diccionario.

json\_readed  # Ahora la base de datos esta en la variable json\_readed

Ahora guardo en el ordenador la base de datos en un archivo llamado data\_json

with open('data.json', 'w+') as outfile:

    json.dump(json\_readed, outfile)

He utilizado el método json.dump de request y me lo ha convertido en el archivo data.json pero todos los datos están en una línea

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Para ponerlo de una manera más legible lo idento, es lo mismo que arriba pero le he puesto la condición indent=4. Lo guardo en un nuevo archivo llamado data\_indented

with open('data\_indented.json', 'w+') as outfile:

    json.dump(json\_readed, outfile, indent=4) #indent=4 es la forma correcta de visualizar un json. Se crea el archivo que he llamado data\_indented.json en nuestra carpeta

Para crear un DataFrame (visualizar los datos en un tabla) puedo hacerlo de dos formas:

1.- Desde el archivo json\_read pues ya tiene estructura de diccionario

df = pd.DataFrame(json\_readed)

df

2.- Desde el archivo identado que he guardado anteriormente en local.

df\_json = pd.read\_json("data\_indented.json")

df\_json

Ahora puede suceder que dentro de alguna de las columnas existan otros json, para ello tendré que repetir el proceso anterior. Compruebo que el tipo de dato de esa columna con type(df\_json[‘nombre columna’] type(df\_json["members"]) y me dira: pandas.core.series.Series

Procedo de la siguiente manera:

1. Leemos data\_indented.json
2. Cargamos el contenido en una variable (members\_json\_list)
3. Cogemos el contenido de la key/columna en el ejemplo "members"

with open("data\_indented.json", "r+") as outfile:

    json\_readed = json.load(outfile)

    members\_json\_list = json\_readed["members"]

members\_json\_list

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Guardo los datos en un fichero (por ejemplo final\_data\_json)

with open('final\_data.json', 'w+') as outfile:

    json.dump(members\_json\_list, outfile, indent=4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cargo este fichero con pandas, (muestro un dataframe)

df\_members\_json = pd.read\_json("final\_data.json")

df\_menbers

y podré visualizar una nueva tabla del diccionario members, donde cada columna son las claves que contenía ese diccionario llamado members.

Si alguna de esas claves del diccionario members contenía a otro diccionario, deberé repetir este proceso.