

```
{}
```

Working on Jsons

```
1] ▶ MI
```

```
import requests
import json
import pandas as pd
```

```
20] ▶ MI
```

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

json_readed = r.json()
json_readed
```

```
{'squadName': 'Super Hero Squad',
 'homeTown': 'Metro City',
 'formed': 2016,
 'secretBase': 'Super tower',
 'active': True,
 'members': [{ 'name': 'Molecule Man',
  'age': 29,
  'secretIdentity': 'Dan Jukes',
  'powers': ['Radiation resistance', 'Turning tiny', 'Radiation blast']},
 {'name': 'Madame Uppercut',
  'age': 39,
  'secretIdentity': 'Jane Wilson',
  'powers': ['Million tonne punch',
  'Damage resistance',
  'Superhuman reflexes']},
 {'name': 'Eternal Flame',
  'age': 1000000,
  'secretIdentity': 'Unknown',
  'powers': ['Immortality',
  'Heat Immunity',
  'Inferno',
  'Teleportation',
  'Interdimensional travel']}]}
```

Esto es un ejemplo para insertar dos puntos ":" en la séptima línea de un hipotético string en el que le falta ese mismo str.

```
-] ▶ MI
```

```
ulr_string = r.text
type(ulr_string)
with open('data_no_json.txt', 'w+') as outfile:
    outfile.write(ulr_string)
```

```
28] ▶ MI
```

```
l = ["a", "b", "c", "d"]
l.insert(2, "X")
l
```

```
['a', 'b', 'X', 'c', 'd']
```

[27] ▶ MI

```
with open('data_no_json.txt', 'r+') as outfile:
    for i in range(7):
        line = outfile.readline()
        print(line)
        l_line = list(line)
        l_line.insert(6, ":")
```

```
{
  "squadName" : "Super Hero Squad",
  "homeTown" : "Metro City",
  "formed" : 2016,
  "secretBase" : "Super tower",
  "active" : true,
  "members" : [
```

[11] ▶ MI

```
print(json_readed)
```

```
{'squadName': 'Super Hero Squad', 'homeTown': 'Metro City', 'formed': 2016, 'secretBase': 'Super tower', 'active': True, 'members': [{'name': 'Molecule Man', 'age': 29, 'secretIdentity': 'Dan Jukes', 'powers': ['Radiation resistance', 'Turning tiny', 'Radiation blast']}, {'name': 'Madame Uppercut', 'age': 39, 'secretIdentity': 'Jane Wilson', 'powers': ['Million tonne punch', 'Damage resistance', 'Superhuman reflexes']}, {'name': 'Eternal Flame', 'age': 1000000, 'secretIdentity': 'Unknown', 'powers': ['Immortality', 'Heat Immunity', 'Inferno', 'Teleportation', 'Interdimensional travel']}]}
```

Save json in a local file called "data.json"

[9] ▶ MI

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

[14] ▶ MI

```
with open('data.json') as f:
    line = f.readline()
    if line:
        print(line)

    line = f.readline()
    print("Esta es la siguiente línea y está vacía:", line)
```

```
{"squadName": "Super Hero Squad", "homeTown": "Metro City", "formed": 2016, "secretBase": "Super tower", "active": true, "members": [{"name": "Molecule Man", "age": 29, "secretIdentity": "Dan Jukes", "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]}, {"name": "Madame Uppercut", "age": 39, "secretIdentity": "Jane Wilson", "powers": ["Million tonne punch", "Damage resistance", "Superhuman reflexes"]}, {"name": "Eternal Flame", "age": 1000000, "secretIdentity": "Unknown", "powers": ["Immortality", "Heat Immunity", "Inferno", "Teleportation", "Interdimensional travel"]}]}
Esta es la siguiente:
```

Save with indent

```
[16] > MI
with open('data_indented.json', 'w+') as outfile:
    json.dump(json_readed, outfile, indent=4)
```

Read local json

```
] > MI
with open('data_indented.json', 'r+') as outfile:
    json_readed = json.load(outfile)
    print(type(json_readed))
    json_readed
```

```
<class 'dict'>
```

```
{'squadName': 'Super Hero Squad',
 'homeTown': 'Metro City',
 'formed': 2016,
 'secretBase': 'Super tower',
 'active': True,
 'members': [{ 'name': 'Molecule Man',
                'age': 29,
                'secretIdentity': 'Dan Jukes',
                'powers': ['Radiation resistance', 'Turning tiny', 'Radiation blast']},
 { 'name': 'Madame Uppercut',
   'age': 39,
   'secretIdentity': 'Jane Wilson',
   'powers': ['Million tonne punch',
              'Damage resistance',
              'Superhuman reflexes']},
 { 'name': 'Eternal Flame',
   'age': 1000000,
   'secretIdentity': 'Unknown',
   'powers': ['Immortality',
              'Heat Immunity',
```

```
[19] > MI
df = pd.DataFrame(json_readed) # From dict
df
```

	squadName	homeTown	formed	secretBase	active	members
0	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Molecule Man', 'age': 29, 'secretIde...
1	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Madame Uppercut', 'age': 39, 'secret...
2	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Eternal Flame', 'age': 1000000, 'sec...

```
[32] > MI
df_json = pd.read_json("data_indented.json")
df_json
```

	squadName	homeTown	formed	secretBase	active	members
0	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Molecule Man', 'age': 29, 'secretIde...
1	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Madame Uppercut', 'age': 39, 'secret...
2	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Eternal Flame', 'age': 1000000, 'sec...

Data Mining & Data Wrangling

As you can see, there are jsons inside the original json. For that, we have to mining the json (data mining) to be able to use it correctly (data wrangling).

How do you solve this issue? Research about this and try a solution.

```
34] ▶ MI  
type(df_json["members"])
```

pandas.core.series.Series

```
38] ▶ MI  
with open('data_indented.json', 'r') as outfile:  
    json_readed = json.load(outfile)  
    for dict_ in json_readed["members"]:  
        print(dict_)  
  
{'name': 'Molecule Man', 'age': 29, 'secretIdentity': 'Dan Jukes', 'powers': ['Radiation resistance', 'Turning tiny', 'Radiation blast']}  
{'name': 'Madame Uppercut', 'age': 39, 'secretIdentity': 'Jane Wilson', 'powers': ['Million tonne punch', 'Damage resistance', 'Superhuman reflexes']}  
{'name': 'Eternal Flame', 'age': 1000000, 'secretIdentity': 'Unknown', 'powers': ['Immortality', 'Heat Immunity', 'Inferno', 'Teleportation', 'Interdimensional travel']}
```

1. Leemos data_indented.json
2. Cargamos el contenido en una variable
3. Cogemos el contenido de la key/columna "members"

```
[40] ▶ MI  
with open("data_indented.json", "r+") as outfile:  
    json_readed = json.load(outfile)  
    members_json_list = json_readed["members"] # Real json  
members_json_list
```

```
[{'name': 'Molecule Man',  
  'age': 29,  
  'secretIdentity': 'Dan Jukes',  
  'powers': ['Radiation resistance', 'Turning tiny', 'Radiation blast']},  
 {'name': 'Madame Uppercut',  
  'age': 39,  
  'secretIdentity': 'Jane Wilson',  
  'powers': ['Million tonne punch',  
  'Damage resistance',  
  'Superhuman reflexes']},  
 {'name': 'Eternal Flame',  
  'age': 1000000,  
  'secretIdentity': 'Unknown',  
  'powers': ['Immortality',  
  'Heat Immunity',  
  'Inferno',  
  'Teleportation',  
  'Interdimensional travel']}]
```

1. Guardamos los datos en un fichero "final_data.json"
2. Cargamos el fichero con pandas

```
41] ► MI
with open('final_data.json', 'w+') as outfile:
    json.dump(members_json_list, outfile, indent=4)
```

```
43] ► MI
df_members_json = pd.read_json("final_data.json")
df_members_json
```

	name	age	secretIdentity	powers
0	Molecule Man	29	Dan Jukes	[Radiation resistance, Turning tiny, Radiation...
1	Madame Uppercut	39	Jane Wilson	[Million tonne punch, Damage resistance, Super...
2	Eternal Flame	1000000	Unknown	[Immortality, Heat Immunity, Inferno, Teleport...

Nota:

1. Con dumps, cargamos el contenido de un diccionario a formato string
2. Con loads, cargamos el contenido de un string a formato json formal.

Concatenamos por columnas los dataframes

```
[47] ► MI
final_df = pd.concat([df_json, df_members_json], axis=1)
final_df

File "", line 1
    final_df = pd.concat([df_json, df_members_json], axis=1)
                                ^
SyntaxError: invalid syntax
```

```
[48] ► MI
final_df = final_df.drop(["members"], axis=1)
final_df
```

	squadName	homeTown	formed	secretBase	active	name	age	secretIdentity	powers
0	Super Hero Squad	Metro City	2016	Super tower	True	Molecule Man	29	Dan Jukes	[Radiation resistance, Turning tiny, Radiation...
1	Super Hero Squad	Metro City	2016	Super tower	True	Madame Uppercut	39	Jane Wilson	[Million tonne punch, Damage resistance, Super...
2	Super Hero Squad	Metro City	2016	Super tower	True	Eternal Flame	1000000	Unknown	[Immortality, Heat Immunity, Inferno, Teleport...