

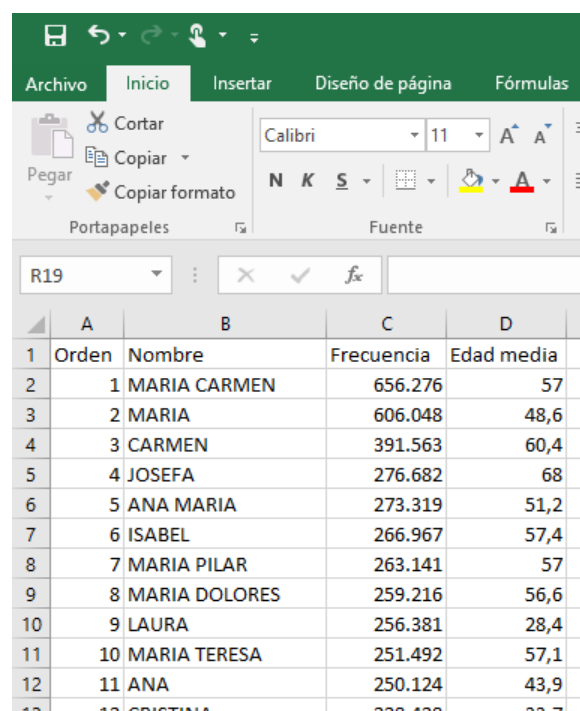
# Archivos CSV y el formato JSON

En Python y en cualquier otro lenguaje, los datos de entrada pueden proceder de distintas fuentes: lo más sencillo es introducir datos desde la consola, también es posible hacer que el programa los descargue de una página web de Internet, o que extraiga la información contenida en un archivo almacenado en el disco.

En este último caso, un archivo puede venir en texto plano, o codificado en un formato especial, como es el caso de los archivos `csv` o los archivos `json`. En esta pequeña sesión atendemos a estos dos casos.

## Archivos CSV y el formato JSON

He aquí un fragmento de un archivo `csv`, abierto con excel:



	A	B	C	D
1	Orden	Nombre	Frecuencia	Edad media
2	1	MARIA CARMEN	656.276	57
3	2	MARIA	606.048	48,6
4	3	CARMEN	391.563	60,4
5	4	JOSEFA	276.682	68
6	5	ANA MARIA	273.319	51,2
7	6	ISABEL	266.967	57,4
8	7	MARIA PILAR	263.141	57
9	8	MARIA DOLORES	259.216	56,6
10	9	LAURA	256.381	28,4
11	10	MARIA TERESA	251.492	57,1
12	11	ANA	250.124	43,9
13	12	CRISTINA	229.129	22,7

El manejo de archivos `csv` es sencillo en Python. Como referencia básica, puede verse la siguiente, entre otras muchas:

<https://docs.python.org/3/library/csv.html>

In [1]:

```
import csv
csvFile = csv.reader(open("nombres_por_edad_media.csv", "r"), delimiter=";")
for row in csvFile:
    print(row)
```

```
['541', 'SARAH', '3.812 ', '19,6']
['542', 'GRACIELA', '3.758 ', '42,5']
['543', 'SACRAMENTO', '3.746 ', '65,5']
['544', 'OIHANE', '3.729 ', '19,9']
['545', 'FERMINA', '3.725 ', '70,9']
['546', 'HAFIDA', '3.706 ', '42,6']
['547', 'PASTORA', '3.702 ', '58,2']
['548', 'SATURNINA', '3.696 ', '74,3']
['549', 'ZAIDA', '3.690 ', '24,8']
['550', 'CELESTINA', '3.689 ', '71,1']
['551', 'MARWA', '3.689 ', '6,8']
['552', 'SERAFINA', '3.683 ', '69,4']
['553', 'MARIA FELISA', '3.653 ', '60,1']
['554', 'PAZ', '3.630 ', '55,0']
['555', 'EVELYN', '3.626 ', '21,9']
['556', 'ISABELLA', '3.625 ', '10,0']
['557', 'NAJAT', '3.621 ', '38,8']
['558', 'SALOME', '3.618 ', '38,6']
['559', 'MARIA SANDRA', '3.616 ', '42,3']
['560', 'LUZ', '3.561 ', '51,6']
```

In [2]:

# De otro modo:

```
import csv
with open('nombres_por_edad_media.csv', 'r') as csvFile:
    reader = csv.reader(csvFile, delimiter=';')
    for row in reader:
        print(row)
```

```
['Orden', 'Nombre', 'Frecuencia', 'Edad media']
['1', 'MARIA CARMEN', '656.276 ', '57,0']
['2', 'MARIA', '606.048 ', '48,6']
['3', 'CARMEN', '391.563 ', '60,4']
['4', 'JOSEFA', '276.682 ', '68,0']
['5', 'ANA MARIA', '273.319 ', '51,2']
['6', 'ISABEL', '266.967 ', '57,4']
['7', 'MARIA PILAR', '263.141 ', '57,0']
['8', 'MARIA DOLORES', '259.216 ', '56,6']
['9', 'LAURA', '256.381 ', '28,4']
['10', 'MARIA TERESA', '251.492 ', '57,1']
['11', 'ANA', '250.124 ', '43,9']
['12', 'CRISTINA', '228.428 ', '33,7']
['13', 'MARIA ANGELES', '226.047 ', '55,4']
['14', 'MARTA', '225.323 ', '29,3']
['15', 'FRANCISCA', '213.820 ', '64,9']
['16', 'ANTONIA', '207.597 ', '64,7']
['17', 'MARIA ISABEL', '204.354 ', '52,8']
['18', 'MARIA JOSE', '203.283 ', '46,1']
```

In [3]:



```
# Escritura:

import csv

ids_columnas = ['Nombre', 'Matemáticas', 'Lengua', 'Historia']

filas = [ ['Juan', '5.7', '2.5', '9.0'],
          ['Sara', '9.5', '6.7', '9.3'],
          ['Alberto', '7.5', '7.5', '7.5'],
          ['Sara', '4.9', '5.2', '8.0']]

id_archivo = "calificaciones_2019.csv"

with open(id_archivo, 'w', newline='') as csv_archivo:
    csvwriter = csv.writer(csv_archivo, delimiter=";")
    csvwriter.writerow(ids_columnas)
    csvwriter.writerows(filas)

print("Hecho")
```

Hecho

## Transformación de datos

Los datos de un csv son siempre cadenas de caracteres, pero se pueden convertir en los formatos necesarios con las funciones (y librerías) adecuadas:

In [4]:



```
#Enteros:
print(int("7"),int("123.000".replace('.', '')))

#Reales:
print(float("4.5"), float("4,5".replace(",",".")))
print(float("123.000,75".replace('.', '').replace(',','.')))

from datetime import datetime
fecha_str = '10-24-2019'

fecha_objeto = datetime.strptime(fecha_str, '%m-%d-%Y').date()
print(type(fecha_objeto))
print(fecha_objeto)
```

```
7 123000
4.5 4.5
123000.75
<class 'datetime.date'>
2019-10-24
```

## El formato JSON

El formato `json` es una notación sencilla para especificar datos y facilitar su intercambio. En la wikipedia puede leerse que se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, actualmente se considera (año 2019) un formato independiente

Se emplea a menudo como alternativa a XML, especialmente en contextos (como E9) en formato independiente del lenguaje.

La idea subyacente a este formato es explotar la codificación mediante el emparejamiento de clave-valor, y la utilización de listas. Los siguientes ejemplos se han tomado de la dirección siguiente:

<https://support.oneskyapp.com/hc/en-us/articles/208047697-JSON-sample-files>

Ejemplo 1:

```
{  
  "fruit": "Apple",  
  "size": "Large",  
  "color": "Red"  
}
```

Ejemplo 2:

```
{
  "quiz": {
    "sport": {
      "q1": {
        "question": "Which one is correct team name in NBA?",
        "options": [
          "New York Bulls",
          "Los Angeles Kings",
          "Golden State Warriros",
          "Huston Rocket"
        ],
        "answer": "Huston Rocket"
      }
    },
    "maths": {
      "q1": {
        "question": "5 + 7 = ?",
        "options": [
          "10",
          "11",
          "12",
          "13"
        ],
        "answer": "12"
      },
      "q2": {
        "question": "12 - 8 = ?",
        "options": [
          "1",
          "2",
          "3",
          "4"
        ],
        "answer": "4"
      }
    }
  }
}
```

El emparejamiento de clave-valor nos recuerda los diccionarios; las listas, las listas.

Trabajemos con dos archivos cuyos contenidos son los mostrados en los ejemplos anteriores:

In [5]:



```
import json

archivo = open("example_1.json")
datos = json.loads(archivo.read())
archivo.close()
print(datos)
```

```
{'fruit': 'Apple', 'size': 'Large', 'color': 'Red'}
```

In [6]:



```
with open("example_2.json") as archivo:
    datos = json.loads(archivo.read())
datos
```

Out[6]:

```
{'quiz': {'sport': {'q1': {'question': 'Which one is correct team name in NB
A?',
    'options': ['New York Bulls',
    'Los Angeles Kings',
    'Golden State Warriros',
    'Huston Rocket'],
    'answer': 'Huston Rocket'}}},
'maths': {'q1': {'question': '5 + 7 = ?',
    'options': ['10', '11', '12', '13'],
    'answer': '12'},
'q2': {'question': '12 - 8 = ?',
    'options': ['1', '2', '3', '4'],
    'answer': '4'}}}}
```

In [7]:



```
# Escritura:

with open("example_3.json", "w") as archivo:
    archivo.write(json.dumps(datos))

# Obviamente, los archivos example_2.json y example_3.json son iguales
```

Se puede cargar un archivo json directamente en un dataframe de pandas:

In [8]:



```
import pandas
datos_pandas = pandas.read_json("example_2.json")
print(datos_pandas)
```

```
              quiz
maths  {'q1': {'question': '5 + 7 = ?', 'options': ['...
sport  {'q1': {'question': 'Which one is correct team...
```

In [9]:

```
print(datos_pandas["quiz"])
print("-----")
print(datos_pandas["quiz"]["maths"])
print("-----")
print(datos_pandas["quiz"]["maths"]["q1"])
print("-----")
print(datos_pandas["quiz"]["maths"]["q1"]["options"])
print("-----")
print(datos_pandas["quiz"]["maths"]["q1"]["options"][3])
```

```
maths    {'q1': {'question': '5 + 7 = ?', 'options': ['...
sport    {'q1': {'question': 'Which one is correct team...
Name: quiz, dtype: object
-----
{'q1': {'question': '5 + 7 = ?', 'options': ['10', '11', '12', '13'], 'answer': '12'}, 'q2': {'question': '12 - 8 = ?', 'options': ['1', '2', '3', '4'], 'answer': '4'}}
-----
{'question': '5 + 7 = ?', 'options': ['10', '11', '12', '13'], 'answer': '12'}
-----
['10', '11', '12', '13']
-----
13
```

In [10]:

```
# También se puede manejar la orientación:

estaciones = pandas.read_json("estaciones.json", orient="index")
estaciones
```

Out[10]:

	0	1	2	3	4	5	6	
<b>altitud</b>	98	58	50	80	230	685	100	
<b>indicativo</b>	1387E	1387	1393	1351	1400	1437O	1473A	
<b>indsinop</b>	08002	08001	08006	08004	08040	08043	08039	
<b>latitud</b>	431825N	432157N	430938N	434710N	425529N	424314N	424418N	
<b>longitud</b>	082219W	082517W	091239W	074105W	091729W	085524W	083738W	
<b>nombre</b>	A CORUÑA AEROPUERTO	A CORUÑA	CABO VILAN	ESTACA DE BARES	FISTERRA	MONTE IROITE	PADRÓN	SAN COM AERC
<b>provincia</b>	A CORUÑA	A CORUÑA	A CORUÑA	A CORUÑA	A CORUÑA	A CORUÑA	A CORUÑA	A

7 rows × 291 columns

<  >

In [11]:

```
estaciones = pandas.read_json("estaciones.json", orient="column")
estaciones
```

Out[11]:

	altitud	indicativo	indsinop	latitud	longitud	nombre	provincia
0	98	1387E	08002	431825N	082219W	A CORUÑA AEROPUERTO	A CORUÑA
1	58	1387	08001	432157N	082517W	A CORUÑA	A CORUÑA
2	50	1393	08006	430938N	091239W	CABO VILAN	A CORUÑA
3	80	1351	08004	434710N	074105W	ESTACA DE BARES	A CORUÑA
4	230	1400	08040	425529N	091729W	FISTERRA	A CORUÑA
5	685	1437O	08043	424314N	085524W	MONTE IROITE	A CORUÑA
6	100	1473A	08039	424418N	083738W	PADRÓN	A CORUÑA
7	370	1428	08042	425317N	082438W	SANTIAGO DE COMPOSTELA AEROPUERTO	A CORUÑA

In [12]:

```
estaciones.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 291 entries, 0 to 290
Data columns (total 7 columns):
altitud      291 non-null int64
indicativo   291 non-null object
indsinop     291 non-null object
latitud      291 non-null object
longitud     291 non-null object
nombre       291 non-null object
provincia    291 non-null object
dtypes: int64(1), object(6)
memory usage: 16.0+ KB
```