



PYTHON

File Processing
CSV file reading and writing

FILE PROCESSING

CSV file reading and writing

FILE PROCESSING

- CSV (*Comma Separated Values*) es un popular formato de archivo utilizado para el intercambio de datos entre aplicaciones.
- Un documento CSV es un archivo de texto con la extensión **.csv** formado por líneas (registros) cuyos campos se encuentran (habitualmente) separados por comas, aunque admite otros separadores como el punto y coma o el tabulador.
- La primera línea puede contener un encabezado.
- Ejemplo. Fichero **contacts.csv**

```
Name,Phone  
mother,222-555-101  
father,222-555-102  
wife,222-555-103  
mother-in-law,222-555-104
```

FILE PROCESSING

- Módulo **csv**
 - Módulo de la biblioteca estándar.
 - Permite leer (**objeto reader**) y escribir (**objeto writer**) datos en formato CSV.
 - La **función reader** recibe la referencia al fichero abierto (el parámetro **newline** es útil para evitar conflictos con las diferentes representaciones de los saltos de línea) y el delimitador. Proporciona un **objeto reader**
 - El objeto **reader** es un elemento iterable con las líneas del fichero. Cada línea se representa como una **lista**.

```
import csv

with open('contacts.csv', newline='') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    for row in reader:
        print(row)
```



```
['Name', 'Phone']
['mother', '222-555-101']
['father', '222-555-102']
['wife', '222-555-103']
['mother-in-law', '222-555-104']
```

FILE PROCESSING

- Módulo **csv**

- La clase **DictReader** actúa como **wrapper** convirtiendo el fichero csv a una estructura de datos en la que cada línea es un diccionario en el que las claves son los títulos de las columnas.
- Las discrepancias entre los nombres de las claves y los títulos de las columnas provocan un **KeyError**.

```
import csv

with open('contacts.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        print(row['Name'], ':', row['Phone'])
```

FILE PROCESSING

- Módulo **csv**
 - Si el archivo utilizado como argumento de **DictReader** no tiene encabezados, se pueden indicar con el parámetro **fieldnames**.
 - Nota: si se indican las cabeceras a través de **fieldnames** y el fichero tiene cabeceras, estas se tomarán como un registro cualquiera.

```
import csv

with open('contacts.csv', newline='') as csvfile:
    fieldnames = ['Name', 'Phone']
    reader = csv.DictReader(csvfile, fieldnames=fieldnames)
    for row in reader:
        print(row['Name'], row['Phone'])
```

FILE PROCESSING

- Módulo **csv**. Almacenamiento de ficheros.
 - Se realiza a través del **objeto writer** proporcionado por la **función writer**.
 - Los métodos **writerow** y **writerows** permiten escribir datos en el fichero.
 - Los datos se indican como un iterable (lista o tupla, principalmente).
 - El fichero de destino debe abrirse en modo escritura.

```
import csv

with open('exported_contacts.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=',')

    writer.writerow(['Name', 'Phone'])
    writer.writerow(['mother', '222-555-101'])
    writer.writerow(['father', '222-555-102'])
    writer.writerow(['wife', '222-555-103'])
    writer.writerow(['mother-in-law', '222-555-104'])
```

FILE PROCESSING

quotechar y quoting pueden utilizarse también en la lectura

- Módulo **csv**. Manejo de comillas.
 - Si no se indica lo contrario, sólo se entrecomillan los campos que contienen el carácter utilizado con delimitador. Se utilizan por defecto las **comillas dobles**.
 - Se puede cambiar el carácter utilizado para entrecomillar con el parámetro **quotechar** (sólo debe asignarse un carácter).
 - El parámetro **quoting** determina qué debe entrecomillarse.
 - **csv.QUOTE_MINIMAL** → Sólo los conflictos con los delimitadores.
 - **csv.QUOTE_ALL** → Todo.
 - **csv.QUOTE_NONNUMERIC** → Sólo valores no numéricos.
 - **csv.QUOTE_STRINGS** → Sólo cadenas de caracteres
 - **csv.QUOTE_NONE** → No entrecomilla nada. En caso de conflicto con el separador → ERROR.

```
import csv
```

```
with open('exported_contacts.csv', 'w', newline='') as csvfile:  
    writer = csv.writer(csvfile, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)  
  
    writer.writerow(['Name', 'Phone'])  
    writer.writerow(['mother', '222-555-101'])  
    writer.writerow(['father', '222-555-102'])
```


FILE PROCESSING

- Módulo **csv**. Escritura con **DictWriter**.
 - Permite escribir **diccionarios**.
 - Los encabezados son **obligatorios**.

```
import csv

with open('exported_contacts_dict_writer.csv', 'w', newline='') as csvfile:
    fieldnames = ['Name', 'Phone']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerow({'Name': 'mother', 'Phone': '222-555-101'})
    writer.writerow({'Name': 'father', 'Phone': '222-555-102'})
    writer.writerow({'Name': 'wife', 'Phone': '222-555-103'})
    writer.writerow({'Name': 'mother-in-law', 'Phone': '222-555-104'})
    writer.writerow({'Name': 'grandmother, grandfather and auntie', 'Phone': '222-555-105'})
```