

LENGUAJES y HERRAMIENTA PARA CIENCIAS DE DATOS I

Librería estándar de Python IV Módulo csv



Leer un fichero csv

◦ Función `reader()`

◆ Argumentos

- Fuente de datos
- Tipo de separador usado

◆ Iterador para procesar las líneas

◆ Cada línea es una lista

```
1 import csv
2 with open('iris.csv', 'r') as f:
3     reader = csv.reader(f)
4     for row in reader:
5         print(row)
```

Escribir un fichero csv

- Función *writer()*
 - ◆ Crear un objeto para escribir
- Método *writerow()*
 - ◆ Escribir cada fila del fichero

```
1 import csv
2 with open('datos.csv', 'w') as f:
3     writer = csv.writer(f)
4     writer.writerow(('Title 1', 'Title 2', 'Title 3'))
5     for i in range(3):
6         row = ('Fila', i, 'clase')
7         writer.writerow(row)
```

Entrecomillar

- Argumento `quoting` al crear el objeto `writer`
 - ◆ `QUOTE_ALL`
 - ◆ `QUOTE_MINIMAL`
 - ◆ `QUOTE_NONNUMERIC`
 - ◆ `QUOTE_NONE`

Dialectos

- Controlar cómo csv analiza o escribe datos
 - ◆ `list_dialects()`
- Crear un dialecto
 - ◆ `register_dialect(nombre, parametros = valor)`

```
1 import csv
2
3 csv.register_dialect('pipes', delimiter='|')
4
5 with open('testdata.pipes', 'r') as f:
6     reader = csv.reader(f, dialect='pipes')
7     for row in reader:
8         print(row)
```

Usar nombres de campo

- Clases DictReader y DictWriter
 - ◆ Leen y almacenan la información en pares clave:valor
 - ◆ Cada fila un diccionario

```
1 import csv
2 with open('iris.csv', 'r') as f:
3     reader = csv.DictReader(f)
4     for row in reader:
5         print(row)
```

