



EBook Gratis

APRENDIZAJE

CSV

Free unaffiliated eBook created from
Stack Overflow contributors.

#CSV

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con csv	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Procesamiento de archivos csv en varios sistemas e idiomas.....	2
Capítulo 2: Procesamiento de archivos CSV en varios idiomas	4
Introducción.....	4
Examples.....	4
Leer y escribir en Python.....	4
Leer y escribir en Java.....	4
Creditos	7

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [csv](#)

It is an unofficial and free csv ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official csv.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con csv

Observaciones

csv es un acrónimo que significa valores separados por comas. Un archivo csv es texto simple, sin embargo, los caracteres pueden codificarse de varias formas. Cada línea en tal archivo representa un solo *registro*. El contenido de un archivo csv debe entenderse como el contenido de una tabla en la que cada registro representa una fila de las tablas. Aunque el acrónimo indica que los valores en las líneas están separados por comas, de hecho, pueden estar separados por cualquier carácter, y los valores de cadena pueden escaparse, por ejemplo, por apóstrofes o comillas dobles. A menudo, la primera línea de un archivo csv está formateada de la misma forma que las líneas subsiguientes, pero contiene identificadores para los datos que contienen; en estos archivos, esta primera línea proporciona encabezados para la tabla representada por el archivo.

Los archivos csv son muy utilizados para almacenar y transmitir datos. La mayoría de los lenguajes y sistemas completos incluyen, o proporcionan acceso a, medios para analizar líneas dentro de dichos archivos para sus valores.

Examples

Instalación o configuración

Un archivo CSV en sí no requiere instalación ya que es solo un archivo de texto plano, generalmente con la extensión `.csv`.

Un archivo CSV generalmente contiene registros. Cada línea representa un registro y está separada por un delimitador, generalmente una coma; pero los puntos y comas y pestañas también se utilizan con frecuencia. Cada línea debe tener el mismo número de campos.

example.csv:

```
Name, Age, Skill, Height, Friendly
Bob, 22, 10, 6.3, yes
Frank, 12, 7, 5.5, no
```

Procesamiento de archivos csv en varios sistemas e idiomas.

Los archivos csv pequeños y simples pueden construirse usando solo un editor de texto, porque un archivo CSV es simplemente texto. Si tiene un software de hoja de cálculo disponible, generalmente es una forma fácil de abrir y guardar archivos CSV.

Leerlos y escribirlos o, de lo contrario, procesar sus contenidos se hace de manera más eficiente utilizando los productos disponibles para el idioma o los sistemas de su elección.

- Ver [entrar descripción del enlace aquí](#)

Lea Empezando con csv en línea: <https://riptutorial.com/es/csv/topic/7030/empezando-con-csv>

Capítulo 2: Procesamiento de archivos CSV en varios idiomas.

Introducción

Los archivos csv pequeños y simples pueden construirse usando solo un editor de texto. Leerlos y escribirlos o, de lo contrario, procesar sus contenidos se hace de manera más eficiente utilizando los productos disponibles para el idioma o los sistemas de su elección.

Examples

Leer y escribir en Python

CSV (valores separados por comas) es un formato de archivo simple que se utiliza para almacenar datos tabulares, como una hoja de cálculo. Este es un ejemplo mínimo de cómo escribir y leer datos en Python.

Escribir datos en un archivo CSV:

```
import csv
data = [{"Ravi", "9", "550"}, {"Joe", "8", "500"}, {"Brian", "9", "520"}]
with open('students.csv', 'wb') as csvfile:
    writer = csv.writer(csvfile, delimiter=',')
    writer.writerows(data)
```

Leyendo datos de un archivo CSV:

```
import csv
with open('students.csv', 'rb') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=',')
    for row in spamreader:
        print("Name: {} class: {} Marks: {}".format(row[0], row[1], row[2]))
output:
Name: Ravi class: 9 Marks: 550
Name: Joe class: 8 Marks: 500
Name: Brian class: 9 Marks: 520
```

Leer y escribir en Java

El siguiente ejemplo muestra formas de leer y escribir archivos csv sin bibliotecas de terceros.

Escribir CSV

```
public void writeToCsvFile(List<String[]> thingsToWrite, String separator, String fileName){
    try (FileWriter writer = new FileWriter(fileName)){
        for (String[] strings : thingsToWrite) {
            for (int i = 0; i < strings.length; i++) {
```

```

        writer.append(strings[i]);
        if(i < (strings.length-1))
            writer.append(separator);
    }
    writer.append(System.lineSeparator());
}
writer.flush();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

Leer CSV

```

// Allows to define custom separator
public List<String[]> readFromCsvFile(String separator, String fileName){
    try (BufferedReader reader = new BufferedReader(new FileReader(fileName))){
        List<String[]> list = new ArrayList<>();
        String line = "";
        while((line = reader.readLine()) != null){
            String[] array = line.split(separator);
            list.add(array);
        }
        return list;
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}

```

También hay algunas bibliotecas de terceros precompiladas que proporcionan formas convenientes de analizar archivos csv. A continuación se presentan algunos ejemplos de tales bibliotecas.

OpenCSV

OpenCSV se considera muy simple de usar y proporciona funcionalidades flexibles al analizar archivos CSV

```

/** Reading CSV */
// Allows varied parameters through constructors to define quote character, number of lines to skip, etc.
try(CSVReader reader = new CSVReader(new FileReader("yourfile.csv"), separator)){
    List<String[]> = reader.readAll();
    // Do something with the data
}

/** Writing CSV */
List<String[]> listToWrite= //fetch the list of string array to write;
try(CSVWriter writer = new CSVWriter(new FileWriter(fileName), separator)){
    writer.writeAll(listToWrite);
    writer.flush();
}

/** Dumping database records to CSV */
// Initialize CSVWriter and fetch resultSet from database ...
writer.writeAll(resultSet, includeColumnNames);

```

OpenCSV también permite enlazar los registros directamente a JavaBeans. Para más información consulte la documentación oficial [aquí](#) .

Otras bibliotecas conocidas incluyen [SuperCSV](#) y [CommonsCSV](#) que proporcionan algunas funcionalidades avanzadas también. Consulte la documentación oficial para más información.

Lea [Procesamiento de archivos CSV en varios idiomas](#). en línea:

<https://riptutorial.com/es/csv/topic/8862/procesamiento-de-archivos-csv-en-varios-idiomas->

Creditos

S. No	Capítulos	Contributors
1	Empezando con csv	Action Dan , Bill Bell , Community , Danny_ds , depperm
2	Procesamiento de archivos CSV en varios idiomas.	Danny_ds , ravigadila , Setu