LEARNING CSV

#csv

Table of Contents

About ... 1

Chapter 1: Getting started with csv ... 2

- Remarks ... 2
- Examples ... 2
  - Installation or Setup ... 2
  - Processing csv files in various systems and languages ... 2

Chapter 2: CSV file processing in various languages ... 3

- Introduction ... 3
- Examples ... 3
  - Reading and writing in Python ... 3
  - Reading and writing in Java ... 3

Credits ... 6

About

Chapter 1: Getting started with csv

Remarks

csv is an acronym that stands for comma-separated values. A csv file is plain text however the characters may be encoded in any of a variety of ways. Each line in such a file represents a single *record*. A csv file's contents are to be understood as the contents of a table in which each record represents one row of the tables. Although the acronym indicates that the values in lines are separated by commas they may, in fact, be separated by any character, and string values may be escaped by, for instance, apostrophes or double quotation marks. Often the first line of a csv file is formatted is the same way as subsequent lines but contains identifiers for the data in them; in these files this first line provides headers for the table represented by the file.

csv files are heavily used for storing and transmitting data. Most full-blown languages and systems include, or provide access to, means for parsing lines within such files for their values.

Examples

Installation or Setup

A CSV file itself requires no installation as it is just a plain text file, usually with `.csv` extension.

A CSV file usually contains records. Each line represents one record and is separated by a delimiter, most commonly a comma; but semicolons and tabs are also frequently used. Each line should have the same number of fields.

example.csv:

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

Processing csv files in various systems and languages

Small, simple csv files can be built using just a text editor, because a CSV file is simply text. If you have spreadsheet software available these are usually an easy way to open and save CSV files.

Reading and writing them, or otherwise processing their contents is done more efficiently using the products available for one's language or systems of choice.

- See enter link description here

Read Getting started with csv online: https://riptutorial.com/csv/topic/7030/getting-started-with-csv

Chapter 2: CSV file processing in various languages

Introduction

Small, simple csv files can be built using just a text editor. Reading and writing them, or otherwise processing their contents is done more efficiently using the products available for one's language or systems of choice.

Examples

Reading and writing in Python

CSV(Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet. This is a minimal example of how to write & read data in Python.

Writing data to a CSV file:

```
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)
```

Reading data from a CSV file:

```
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
```

Reading and writing in Java

Below example shows ways to read and write csv file without any third party libraries.

Write 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();
        }
}
```

Read 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;
    }
}
```

There are also some pre-compiled third party libraries which provide convenient ways to parse csv files. Below are some examples of such libraries.

OpenCSV

OpenCSV is considered very simple to use and provides flexible functionalities while parsing CSV files

```
/** 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 also allowes binding the records directly to JavaBeans. for more information refer official documentation here.

Other known libraries include SuperCSV and CommonsCSV which provide some advanced functionalities as-well. Refer to official documentation for more information.

Read CSV file processing in various languages online: https://riptutorial.com/csv/topic/8862/csv-file-processing-in-various-languages

Credits

S. No	Chapters	Contributors
1	Getting started with csv	Action Dan, Bill Bell, Community, Danny_ds, depperm
2	CSV file processing in various languages	Danny_ds, ravigadila, Setu