**跨平台的AliceCSV解析软件**

**使用手册**

版本：V1.0

**目录**

[一、软件概述 3](#_Toc173189934)

[二、功能介绍与使用方法 4](#_Toc173189935)

[1. 解析CSV文件内容为二维列表 4](#_Toc173189936)

[2. 解析CSV文件的其中一行 4](#_Toc173189937)

[3. 将表格输出为CSV文件 4](#_Toc173189938)

[4. 修复CSV文件中出现的长度问题 5](#_Toc173189939)

[5. 格式转换 6](#_Toc173189940)

# 一、软件概述

跨平台的AliceCSV解析软件是一款跨平台的对CSV文件进行操作的软件，可以把CSV解析为通用的二维列表或数组，简化了软件开发中对CSV文件操作的过程。软件还包含对CSV文件进行格式纠正和格式转换的功能。

软件具有兼容不规范的CSV文件的能力，针对用户在操作CSV文件时常见的错误进行了优化，在发现文件包含格式错误时能尽量还原作者原本的意图。

软件包含了C++、Python和JavaScript语言的实现，分别对应嵌入式软件和应用程序、数据处理和网页前端用途，可覆盖大多数开发用途。多种实现使得软件可以跨平台使用。

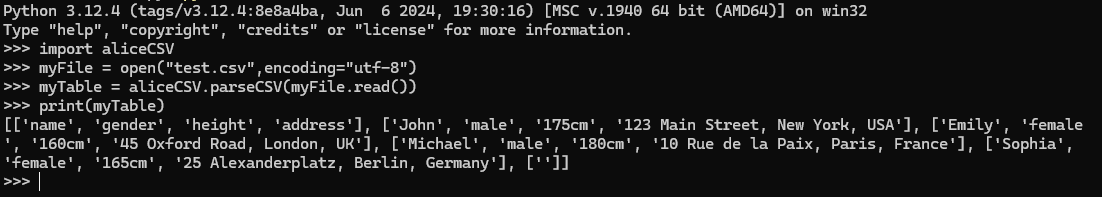
相比同类软件，跨平台的AliceCSV解析软件非常简洁，其中C++实现除去注释和示例后全部代码仅4476字节，便提供了CSV文件的解析、转换等功能，有效减少了处理CSV文件所需的硬件资源，可用于嵌入式设备。

# 二、功能介绍与使用方法

### 1. 解析CSV文件内容为二维列表

本软件提供的parseCSV函数用于解析CSV文件的内容。

以Python版本为例，读取解析的CSV文件的文本内容，传入parseCSV函数，即可得到解析为二维列表的表格。



函数需要的参数和对应的含义如下：

parseCSV(csv\_text, [可选]delimiter)

csv\_text：要解析的CSV文件文本。

delimiter : CSV文件的分隔符，可以不填，默认为","。

### 2. 解析CSV文件的其中一行

用户可使用软件的parseLine函数来解析CSV文件的单独某一行。

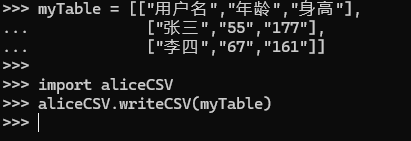
parseLine(line, delimiter):

line：CSV文件某一行的文本。

delimiter：解析时选择的分隔符。可以不填，默认为”,”。

### 3. 将表格输出为CSV文件

writeCSV函数可以将二维列表表示的表格保存为CSV文件。





**注意，由于各编程语言的IO逻辑不同，不同实现存在一定差异。例如，JavaScript实现的writeCSV函数返回的是一个blob形式的CSV文件，而不是直接写入到硬盘。**

函数需要的参数和对应的含义如下：

writeCSV(sheet, [可选]output\_path, [可选]delimiter,

[可选]sheet\_encoding, [可选]line\_break)

sheet：需要保存的二维列表。

output\_path: 输出路径。可以不填，默认为在当前目录下创建"output.csv"。

sheet\_encoding: 输出文件的编码格式。可以不填，默认为“utf-8"。

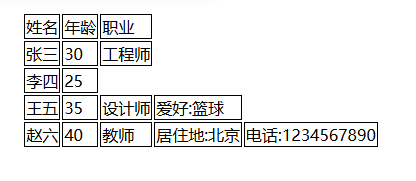
delimiter: CSV文件所使用的分隔符。可以不填，默认为","。

line\_break: 输出文件使用的换行符。可以不填，默认为"\n"。

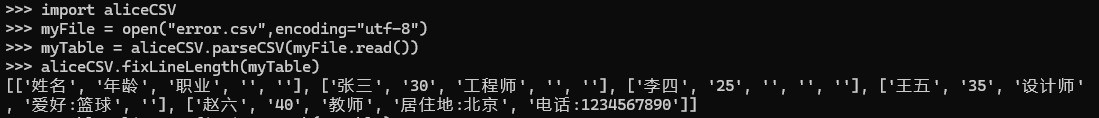
### 4. 修复CSV文件中出现的长度问题

由于种种原因，一些CSV文件每行的字段数不一样，这不符合国际上常用的RFC 4180规范，在一些场景下可能会引发问题。用户可以使用软件的fixLineLength函数使得每行字段数相同。

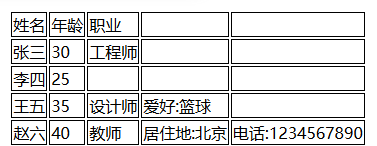
例如，下面这个表格各行长度不同。



我们可以使用fixLineLength进行修复



把结果保存为CSV文件再打开，可以看到表格的每一行长度都变为了5。



函数需要的参数和对应的含义如下：

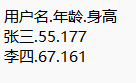
fixLineLength (csv\_sheet)

csv\_sheet：二维列表表示的表格。

### 5. 格式转换

软件的fixCSV函数可以把CSV文件保存为各种兼容格式的CSV文件，包括改为使用其他分隔符、文件编码、换行符等。

例如，对于一个分隔符是”.”的CSV文件，可以使用fixCSV函数把它转换为常见的以逗号为分隔符的CSV文件。



如图，使用Python实现的fixCSV函数，输入源文件地址和源文件分隔符，即可在当前路径输出转换成的“output.csv”。

文本

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

**注意，由于各编程语言IO操作的逻辑不同，不同实现会有一定差异。在JavaScript实现中，fixCSV函数返回的是一个Promise，用户解析这个Promise即可得到blob形式的转换后的文件。**

可以根据实际需要输入更多参数。函数需要的参数和对应的含义如下：

fixCSV(path, [可选]output\_path,[可选]origin\_delimiter,

[可选]target\_delimiter, [可选]origin\_encoding)

path: 输入的初始CSV文件路径.

output\_path: 输出生成的CSV文件的路径。可以不填，默认为“output.csv”。JavaScript实现没有这一参数。

origin\_delimiter：原始CSV文件的分隔符。可以省略，默认值为“,”。

target\_delimiter：输出文件中使用的分隔符。可以省略，默认值为“,”。

origin\_encoding：原始文件的编码。可以省略，默认值为“utf-8”。

target\_encoding：输出文件中使用的编码。可以省略，默认值为“utf-8”。

target\_line\_break：输出文件的换行符。可以省略，默认值为“\n”。