

# tabular2 宏包示例文档

Ms\_yam

2025 年 9 月 3 日

## 摘要

本文档是 tabular2 宏包文档的扩展文档，展示了 tabular2 的一些基本功能及用法。同时，本文档也是 tabular2 的功能测试文档。文档本身只作功能介绍及效果展示，并未同时显示生成表格的相关代码，需要用户自行查看源代码<sup>1</sup>。查看源代码之前，应当先阅读 tabular2 宏包文档的用户部分。

## 目录

<b>1 行列名设置</b>	<b>2</b>
<b>2 data 环境录入数据</b>	<b>2</b>
2.1 标准输入	2
2.1.1 默认效果	2
2.1.2 特殊字符	3
2.2 CSV 输入	3
2.2.1 默认效果	3
2.2.2 特殊字符【待处理】	4
2.3 JSON 输入	4
2.3.1 默认效果	4
2.3.2 特殊字符【待处理】	4
2.4 输入位置	4
2.5 无表头输入	5
2.6 文件导入【待实现】	5
<b>3 行列与单元格输入</b>	<b>5</b>
3.1 按行输入	5
3.2 按列输入	6
3.3 按单元格输入	6

---

<sup>1</sup>现在编辑器和 PDF 阅读器均支持反向查看，要找对应的源代码几乎无难度。

1 行列名设置	2
4 加载与保存	6
4.1 保存表格	6
4.2 加载表格	7
5 格式设置	7
5.1 宽度与高度	7
5.2 对齐方式	8
6 合并单元格【待实现】	8
7 渲染输出【待更新】	8

## 1 行列名设置

行列名用于输入时定位，任一行列均可以有多个列名，但一个行列名只能映射到唯一的行或列。行列名映射是可以修改的，但修改后不会影响已录入数据。

单纯的设置行列名会得到一张空表。其中，`\excelcolname` 命令不影响表的行列数；而 `\rowname` 与 `\colname` 命令则会影响。设置列名会修改对应的表头<sup>2</sup>。下面是第一个示例<sup>3</sup>：

5	B	AA	BB	CC
				data

5	B	AA	BB	CC
data				

注：此时的行名对应关系为  $\langle 2 \rangle = 1$  (无效)， $\langle A \rangle = 2$ ；如果这时输入行坐标 2，它会识别为第 2 行而非第 1 行。对于列名同理，虽然列名  $\langle 5 \rangle = 1$ ，但因其不生效，所以最终修改的是第 5 列。

注：行列名暂时不支持特殊字符，后续是否支持待考量。

## 2 data 环境录入数据

### 2.1 标准输入

#### 2.1.1 默认效果

以下是全部使用默认值（逗号分隔，`\\`换行）的标准输入的效果：

<sup>2</sup>表头为最后设置的有效列名。

<sup>3</sup>以框架及三线表两种方式输出。【下同】

Name	Sex	Age
John	man	18
Leon		20
Lily	woman	21

Name	Sex	Age
John	man	18
Leon		20
Lily	woman	21

其中：首行为表头（兼列名），中间的行会保留，尾随的空行不会。不强制要求所有行的列数相同。

### 2.1.2 特殊字符

标准输入不对特殊字符进行任何预处理，所有输入均为凭据表，直接由 TeX 的默认方式处理。但由于 “\” 用于分隔表格行，因此单元格换行需要使用 “\newline”。如果单元格本身带逗号，则可通过指定其它分隔符来规避。

AA	BB	CC
	a	
A{[#\$	b	”_
B}]%&	c	
d	,\	

AA	BB	CC
	a	
A{[#\$	b	”_
B}]%&	c	
d	,\	

## 2.2 CSV 输入

### 2.2.1 默认效果

需要指定输入选项为 “csv” 才能让本环境按 CSV 格式处理。

AA	BB	CC
A	ab	1
B	cd	2
		3

AA	BB	CC
A	ab	1
B	cd	2
		3

虽然本宏包也支持指定分隔符，但是仍建议遵守约定，使用默认值 “,”。

### 2.2.2 特殊字符【待处理】

## 2.3 JSON 输入

### 2.3.1 默认效果

需要指定输入选项为“json”才能让本环境按 JSON 格式处理。

A	B	C	A	B	C
a	true	1	a	true	1
bc		2	bc		2
d		3	d		3
e	false	4	e	false	4

JSON 的输入方式与标准输入迥异，不需要使用“\”来换行。其属性名为表头（兼列名），属性值为表体。

### 2.3.2 特殊字符【待处理】

## 2.4 输入位置

对于标准输入（含 CSV 格式），可以通过选项来指定输入的起点位置<sup>4</sup>。后续位置则依次往后延。如果未指定，则默认为“{1, 1}”。

<1>	<2>	Name	Sex	Age	<1>	<2>	Name	Sex	Age
		John	man	18			John	man	18
		Leon		20			Leon		20
		Lily	woman	21			Lily	woman	21

如果原有列名映射的位置与现有位置不同，则会更新列名映射。但之前按列名输入的数据不会变。

Sex	Name	Age	Sex	Name	Age
man	John		man	John	
man	John	18	man	John	18
	Leon	20		Leon	20
woman	Lily	21	woman	Lily	21

<sup>4</sup>注：如果某列没有列名，则其表头为“<i>”。

但对于 JSON 格式，则是完全不同的逻辑：如果列名已经存在，则直接填充数据；如果列名不存在，则在指定位置及之后依次添加新列名。

## 2.5 无表头输入

在标准输入与 CSV 输入中，默认第一行为表头（列名），但这可能通过 `header` 选项来修改这一设定。

<1>	<2>	<3>
Name	Sex	Age
John	man	18
Leon		20
Lily	woman	21

<1>	<2>	<3>
Name	Sex	Age
John	man	18
Leon		20
Lily	woman	21

如果该选项只在 `data` 层级设置，那么它的作用只限制为把第 1 行当作数据行处理<sup>5</sup>。如果需要一个无表头的表，则需要 `xtable` 层级中设置<sup>6</sup>。

John	man	18
Leon		20
Lily	woman	21

John	man	18
Leon		20
Lily	woman	21

## 2.6 文件导入 `\file` 【待实现】

`\file` 是的功能与 `data` 的功能几乎一致，除了其是以文件作为输入及换行有特殊处理。

# 3 行列与单元格输入

此本部分是本宏包与传统表格之间的最大差异点之一。

## 3.1 按行输入 `\row`

以下是按行输入的示例，如果某行数据超出现有列数，则会自动添加列（无列名）。

<sup>5</sup>此时如果没有额外设置列名，则会出现如示例所示的默认表头。

<sup>6</sup>此时，`data` 层级默认会继承这个设置，除非手动指定。

Name	Sex	Age	<4>
John	man	18	
Lily		20	
	woman	18	newcol
Jacob	man	21	

Name	Sex	Age	<4>
John	man	18	
Lily		20	
	woman	18	newcol
Jacob	man	21	

### 3.2 按列输入\col

以下是按列输入的示例，如果某列数据超出现有行数，则会自动添加行。

Name	Sex	Age	newcol	<5>
John	man	18	A	1
	woman		B	2
	man			

Name	Sex	Age	newcol	<5>
John	man	18	A	1
	woman		B	2
	man			

### 3.3 按单元格输入\cell

以下是按单元格输入的示例，如果某列数据超出现有行数，则会自动添加行。

Name	Sex	Age	newcol
John	man	18	
			newdata

Name	Sex	Age	newcol
John	man	18	
			newdata

## 4 加载与保存

当存在大量相似的表的时，可以使用本功能与减少录入工作。

### 4.1 保存表格

以下是一个基础表，并使用保存命令保存了部分节点（保存数据没有直接体现）。

Name	Sex	Age
John	man	18
Leon		19
Lily	woman	21

Name	Sex	Age
John	man	18
Leon		19
Lily	woman	21

请注意保存的节点，在保存命令之后做的修改不会影响已经保存的表格。

## 4.2 加载表格

以下是复用的表格数据：

Name	Sex	Age
John	man	18
Leon	man	20
Lily	woman	21

Name	Sex	Age
John	man	18
Leon	man	20
Lily	woman	21

注，加载表格会覆盖之前的设置，因此通常把它放置到第一行。

# 5 格式设置

## 5.1 宽度与高度

默认表格会自动计算所需的尺寸，以包容单元格。

AA	BB	CC	EE
abcde	12	dd	abccba
ab	19		
	20	xy	abc

AA	BB	CC	EE
abcde	12	dd	abccba
	19		
ab	20	xy	abc

但可以通过 `\rowheight` 与 `\colwidth` 来指定其它方式。

AA	BB	CC	EE
abcde	12	dd	abccba
ab	19		
	20	xy	abc

AA	BB	CC	EE
abcde	12	dd	abccba
	19		
ab	20	xy	abc

注：如果宽度设置过小，会导致数据超出单元格。

## 5.2 对齐方式

默认情况，表格是左右居中，上下底对齐的，但可通用`\rowalign` 与 `\colalign` 来指定对齐方式。

AA	BB	CC	EE
abcde	12	dd	abccba
ab	19 20	xy	abc

AA	BB	CC	EE
abcde	12	dd	abccba
ab	19 20	xy	abc

注：对齐 Bug 待修复。

## 6 合并单元格【待实现】

## 7 渲染输出【待更新】