

表格

表格更适合用可视化方式来生成；因为它有两个维度，使用线性的标记语言来生成，自然比较麻烦。

DocBook 生成表格虽然麻烦，但是相对 reStructuredText 等使用空间方位生成的方式，却更容易控制

表格分为两种 **table** 和 **informaltable**，区别在于 **table** 可以有标题，能够被收录到目录中。下面是表格的实例：

表 27.1. 表格实例

表头		
单元格一	单元格二	单元格三
11	12	13
	22	23
31		
		表底

```
<table>❶ <title>表格实例</title>❷ <tgroup cols="3">❸ <thead><row><entry></entry><entry>表头</en
```

- ❶ 说明使用的是何种表格， **table** 还是 **informaltable**
- ❷ 表格标题
- ❸ 定义表格宽度（有多少列），必需！
- ❹ 表头，表格的第一行，用粗体显示，非必需
- ❺ 表底，其实我从来没见过有人用它
- ❻ 表体，表格的主体部分
- ❼ 行，每一行可以拥有的单元格不能超过表格宽度的定义
- ❽ 单元格。 **entry** 元素的内容从该行第一格开始排列
- ❾ 第一格留空的话，要用 `<entry></entry>` 占一个位置
- ❿ 留空的单元格之后没有 有内容的单元格，可以省略 `<entry></entry>`

跨行表格

表 27.2. 跨行表格

跨两行	12	13
	22	23
31	跨三行	33
41		跨两行
51		

```
<table> <title>跨行表格</title> <tgroup cols="3"> <tbody> <row> <entry morerows="1"❶>跨两行
```

- ❶ 使用 `morerows="N"` 合并下方的 N 个单元格；
- ❷ 由于该行第一个单元格已经被合并，所以应在这里出现的 **entry** 元素取消
- ❸ 这里有一个 **entry** 被取消
- ❹ 这里有两个 **entry** 被取消

跨列表格

表 27.3. 跨列表格

第一个单元格		第三个单元格	第四个单元格
第一个单元格	第二个单元格		
第一个单元格	第二个单元格	第三个单元格	第四个单元格

```
<table>  <title>跨列表格</title>  <tgroup cols="4"><colspec colnum="1" colname="1"/>❶<colspec colnum="2" colname="2"/><colspec colnum="3" colname="3"/><colspec colnum="4" colname="4"/><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></tgroup></table>
```

- ❶ 先给每列取个名
- ❷ 列编号
- ❸ 列名。可以任意取，但为了清晰，我用编号作为列名
- ❹ 跨列单元格的起始列，用列名指定
- ❺ 跨列单元格的结束列