1 tabular 环境

1.1 tabular 环境的最基本参数

必须在必选参数中指明每一列的左右对齐方式[1]：l（左对齐）、c（水平居中）、r(右对齐)。

A 左对齐　B 水平居中　C 右对齐
D　　E　　F

以上是 tabular 环境中最基本的要件，可以发现，有几个【代表每一列左右对齐方式的字母】，就说明有几列，环境中的 “&” 符号的数量就在这个基础上减一，在每一行的最后使用 “\\” 符号表示该行结束。

1.2 tabular 环境的可选参数

可选参数 “c” 表示表格中央和文本所在行对齐（默认值）。

A	B	C
D	E	F

可选参数 “t” 表示表格上端和文本所在行对齐。

A	B	C
D	E	F

可选参数 “b” 表示表格下端和文本所在行对齐。

A	B	C
D	E	F

1.3 tabular 环境的其他参数

1.3.1 竖直表线和水平表线

在此基础上，可以增加其他参数来丰富表格形式：在必选参数中使用 “|” 符号来增加竖直表线，“|” 符号可以无限叠加，使用 “\hline” 命令增加水平表线，“\hline” 命令也可以无限叠加。

A	B	C
D	E	F

[1]这里选择 “左右对齐”（以及下文的 “上下对齐”）的称呼方式，是为了避免如 “水平对齐”、“垂直对齐” 这种称呼方式引起的误解，但是 “水平居中” 和 “垂直居中” 却不存在这种问题，因此这两个称呼照旧不变。

竖直表线除了可以由 “|” 命令来绘制，也可以使用 “@{intercolumn sign}” 命令绘制，括号中可以选择各种符号，比如 “-”，打印出来的效果就是在每行相应的列与列之间输出这一符号。如果什么符号都不写，即 “@{}”，效果就等同于完全不设置竖直表线。另外，“@{|}” 的效果并不等同于常用的 “|”，后者绘制从上到下连通的竖直表线，而前者是在每一行的相同位置输出一个 “|” 符号。

A-B|C
D-E|F

如果将上述命令中的 “@” 符号替换为 “!” 符号，即 “!{intercolumn sign}” 命令（该命令由 array 宏包提供），也能达到类似的效果，只不过列与列之间的空隙会拉大，看起来更加美观。

A - B | C
D - E | F

可以使用 “\cline{x-y}” 命令绘制从第 x 列到第 y 列的局部水平表线。

A	B	C
B	E	F

可以使用 makecell 宏包提供的 “\Xhline{}” 命令和 “\Xcline{}{}” 命令调整水平表线的宽度。

A	B	C
D	E	F
G	H	I

可以使用 multirow 宏包提供的 “\multicolumn{n}{cols}{text}” 命令设置跨列内容（类似于 Excel 中的合并同行的多列单元格）。

A	B	C
D		E

结合以上两条命令绘制的表线：

例一

A	B	<-C[2]
D		<-E[3]

例二

A	B	C[4]->
D	E	

可以使用 multirow 宏包提供的 “\multirow{number of rows}{width or *}{text}” 命令设置跨行内容（类似于 Excel 中的合并同列的多行单元格）。

设置合并后的列宽：

A	B	C
D	E	F
G	H	
J	K	L

不设置合并后的列宽，使其自适应宽度：

A	B	C
D	E	F
G	H	
J	K	L

不在命令中删除合并前属于合并范围内的其他内容（红色的 F1），会造成奇怪的效果：

A	B	C
D	E	F
G	H	F1
I	J	K

[3]C 左侧的竖直表线通过 tabular 环境的必选参数设置。

[3]E 左侧的竖直表线通过 “\multicolumn{n}{cols}{text}” 命令的第二个必选参数设置。

[4]C 右侧的竖直表线通过 tabular 环境的必选参数设置。

如果要进行设置跨行兼跨列的操作，则需要将 “\multirow{number of rows}{width or *}{text}” 命令放置在 “\multicolumn{n}{cols}{text}” 命令的内部，即先进行合并行的操作，再进行合并列的操作，如果嵌套的顺序反过来，编译会报错，下面分步演示这一操作。

原表格：

A	B	C
D	E	F

先合并行：

A	B	C
D		F

再合并列（第一步）：

A	B
C	

可以发现，上面这一步打印出的效果很奇怪，这是因为只对第一行进行了合并列的操作，第二行内部出现的一个竖直表线正是 tabular 环境的必选参数绘制的，还需要对第二行进行同样的合并列的操作，但是不输入合并列之后填入的内容，这样，得到的才是最终的正确结果。

合并列（第二步）：

A	B
D	

1.3.2 每一列的上下对齐方式

代表每一列左右对齐方式字母的 “lcr” 可以替换成代表每一列上下对齐方式的字母：“p{}”、“m{}”、“b{}”（后两者由 array 宏包提供），分别代表上对齐、垂直居中、下对齐（吴康隆 P52–53 称为 “竖直居上”、“竖直居中”、“竖直居下”），在这三个字母后面的大括号中需要指定列宽，但是同时文字自动左对齐，无法再选择左右对齐方式（但是可以使用 array 宏包提供的 “>{decl}” 命令和 “<{decl}” 命令进行设置，见下文），效果如下所示：

						Texte	texte	texte
						texte	texte	texte
			Texte	texte	texte	texte	texte	texte
			texte	texte	texte	texte	texte	texte
Texte	texte	texte	texte	texte	texte	texte...		
texte	texte	texte	texte	texte	texte			
texte	texte	texte	texte...					
texte	texte	texte						
texte...								

但是经过检验可以发现，所谓的“上对齐”、“下对齐”，并不是直觉上的靠近表格上端和靠近表格下端，反而“上对齐”的文本靠近表格下端，“下对齐”的文本靠近表格上端，这其实反映出一个很关键的概念性问题：从“对齐”的本义出发，同一列上的文本的左右对齐方式应该相同（这也是 tabular 环境的必选参数中“lcr”设置的效果），同一行上的文本的上下对齐方式应该相同。上述命令的打印效果看似违反直觉，其实是由于同一行上的三列设置了不同的上下对齐方式导致的[5]。在 LaTeX 中，表格的对齐方式只能按照列来统一定义，因此可以为不同的列设置不同的左右对齐方式，却不能为不同的行设置不同的上下对齐方式（或许有相应的宏包可以达到这一效果）。

如果为不同列设置相同的上下对齐方式，这样就能发挥上下对齐真正的作用，不过要注意为每一列的文本设置不同的上下长度，以体现上下对齐的效果：

上对齐：

[5]但是上述命令的打印效果仍然可以反映出同一行上的三种不同的上下对齐方式对应的对齐基线被 LaTeX 放置在了同一条水平线上。

Texte texte texte texte texte texte texte texte texte texte texte texte texte texte texte texte...	Texte texte texte texte texte texte texte texte texte texte...	Texte texte...

下对齐：

		Texte texte...
Texte texte...		
texte texte...	Texte texte texte texte texte texte texte texte texte texte...	

1.3.3 为表格中的单词设置自动换行

先提一个问题：为什么 LaTeX 的原生命令中，只为 tabular 环境设置了一种上下对齐方式，即“上对齐”？最主要的原因是“p{}”命令最初的功能可能并不是设置上下对齐方式，而是**设置列宽**，否则无法解释为什么它没有像左右对齐的三个命令一样不带有参数。以上面的这张表格为例，如果必选参数是用 lcr 来表达的，由于没有列宽的定义，表格不会自动换行，而会在同一行上打印出所有的内容，导致表格超出页面。而如果使用“p{}”命令为每一列设置列宽，就可以保证表格在固定的列宽内自动换行，不会超出页

面。

现在再看单词的问题，上文所说的“自动换行”，其实需要符合三个条件：第一，表格设置了列宽；第二，文本的长度超过了所在列的列宽；第三个条件最容易被忽视，那就是这段文本应该包含两个及两个以上的单词。自动换行是从第二个单词开始的，这包含两种情况：第一，文本另起一行开始第二个单词；第二，文本按照一定的标准（通常是按照提前规定好的的语素界限）从第二个单词内部断开，第二个单词的前面一部分留在第一行，后面一部分另起一行开始，甚至单词过长，列宽过小的时候，可能需要将第二个单词分割成两行以上。因此，“自动换行” 的说法，其实是不包含第一个单词在内的。而表格并不像普通的页面，后者一般不可能在一行上连一个单词都容纳不下，表格的一个单元格完全有可能无法容纳下一个比较长的完整的单词，因此，如果要表格内的文本从第一个单词开始就可以根据列宽进行自动换行，需要进行另外的设置，下面这张表格展示了一个长单词、两个长单词在一个列宽短于该单词长度的表格中的表现情况，左边一列利用“\hspace”命令进行了另外的设置（设置的方式见下文介绍），达到了从第一个单词开始就自动换行的效果，右边一列没有进行另外的设置，因此可以看到第一个单词并没有自动换行，而是从第二个单词开始自动换行。

Super- con- scious- ness	Superconsciousness
	Superconsciousness Super- con- scious- ness

1.3.4 在单元格内手动设置换行

上文介绍的是文本在表格内自动换行，但有的时候我们希望能够控制换行的位置，这可以使用 makecell 宏包的“\makecell” 命令完成。

A1A1A1 A2	B C D1 D2D2D2
E1 E2	F G H

A1A1A1 A2	B C D1 D2D2D2
E1 E2	F G H

1.3.5 为每一列设置统一的格式

可以使用 array 宏包提供的 “>{decl}” 命令和 “<{decl}” 命令（“decl” 是 “declaration” 的缩写）为每一列的内容设置统一的格式，前者放在每一列对应的 lcrpmb 参数之前，后者放在 lcrpmb 参数之后，一般情况下设置前者就足够了，后者是在设置环境格式时需要的。这两条命令使得同一列上不仅可以设置对齐方式，而且可以设置除了对齐方式以外的其他格式。

A	B	*C*
D	E	*F*

也可以使用 array 宏包提供的 “\newcolumntype{name}{definition}” 命令为同一列新定义一个格式命令，不过要求命令形式是和 lcrpmb 一样的单字母形式。

A	B	C
D	E	F

前文提到，如果在 tabular 环境的必选参数中设置了同一列的上下对齐方式，就无法再在必选参数中设置该列的左右对齐方式，但是可以通过 array 宏包提供的 “>{decl}” 命令和 “<{decl}” 命令进行设置。

第一种设置方式：

		Texte texte texte texte texte texte
Texte texte texte texte texte texte texte texte texte texte texte texte texte...	Texte texte texte texte texte texte texte...	texte texte texte texte texte texte texte texte texte texte texte texte texte...

第二种设置方式：

Texte texte texte texte texte texte texte texte texte texte...	Texte texte texte texte texte texte texte...	Texte texte...

1.3.6 分割表头

可以使用 diagbox 宏包的 “\diagbox” 命令分割表头。
分割成两部分：

右边 / 左边	A	B
C	D	E
F	G	H

分割成三部分：

中间 / 右边 / 左边	A	B
C	D	E
F	G	H