

第 3 章 排版数学公式

Donald E. Knuth(高德纳)

2017 年 9 月 26 日

目录

1	$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 宏集	2
2	单个方程	2
2.1	数学模式	3
3	构建数学公式块	3
3.1	希腊字母	3
3.2	指数, 上标, 下标	4
3.3	根式	4
3.4	点	4
3.5	横线	4
3.6	水平花括号	4
3.7	数学重音符号	4
3.8	向量	5
3.9	函数名	5
3.10	取模函数	5
3.11	取模函数	5
3.12	二项式系数和符号堆叠	6
3.13	积分运算符、求和运算符、乘积运算符	6
4	长公式折行	7
5	多行公式	7
5.1	传统命令存在的问题	7
5.2	IEEEeqnarray 环境	9
5.3	IEEEeqnarray 环境的一般用法	9
6	数组和矩阵	10
7	数学模式中的空格	11
7.1	幽灵	12

目 录	2
8 折腾数学字体	12
8.1 粗体符号	12
9 定理, 引理.....	12
9.1 证明和证毕符号	13

1 $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX 宏集

在介绍数学公式排版之前，简单介绍一下 $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX 宏集。 $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX 宏集合是美国数学学会 (American Mathematical Society) 提供的对 \LaTeX 原生的数学公式排版的扩展，其核心是 `amsmath` 宏包，对多行公式的排版提供了有力的支持。此外，`amsfonts` 宏包以及基于它的 `amssymb` 宏包提供了丰富的数学符号；`amsthm` 宏包扩展了 \LaTeX 定理证明格式。在本章，需要在导言区使用 `\usepackage{amsmath}` 命令引入 `amsmath` 宏包。

2 单个方程

数学公式有两个排版方式：其一是与文字混排，称为行内公式；其二是单独列为一行排版，称为行间公式。

行内公式由一对 `$` 符号包裹：

示例：

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $a^2 + b^2 = c^2$

\LaTeX is pronounced as $\tau\epsilon\chi$

100 m³ of water

This comes from my ♥

行间公式放在 `\begin{equation}` 和 `\end{equation}` 之间。可以通过 `\label` 和 `\eqref` 来给公式添加标签和建立引用，用 `\tag` 来给公式指定具体的名字。

示例：

Add a squares and b squared to get c squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2 \tag{1}$$

Einstein says

$$E = mc^2 \tag{2}$$

He didn't say

$$1 + 1 = 3 \tag{dumb}$$

This is a reference to (2)

如果不想给公式编号，用 `equation` 的加星版本 `equation*`；或者把公式放在 `\[` 和 `\]` 之间。

示例：

Add a squares and b squared to get c squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2$$

or you can type less for the same effect:

$$a^2 + b^2 = c^2$$

虽然 `\[` 和 `\]` 很简洁，但是使用时不能像 `equation` 和 `equation*` 那样再有编号和无编号之间切换。

注意排版格式中行内公示 (text style) 和行间公式 (display style) 的区别。

示例:

This is text style: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$. And this is display style:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (3)$$

在排版行间公式, 可以把一些比较高的公式放在 `\smash` 命令里, 让 `LATEX` 忽略这些公式的高度, 使行间距保持不变。

示例:

A $d_{e_{p_{e_r}}}$ mathematical expression followed
by a $h^{i_{g^{h_{e_r}}}}$ expression. As opposed to a
smashed $d_{e_{p_{e_r}}}$ expression followed by a $h^{i_{g^{h_{e_r}}}}$
expression.

2.1 数学模式

当你使用 `$` 开启行内公示输入, 或是使用 `\[` 命令、`equation` 环境时, 你就进入了所谓的数学模式。数学模式相比于文本模式有以下特点:

1. 数学模式中输入的空格全部被忽略。数学符号的间隙默认完全由符号的性质 (关系符号、运算符号等) 决定。需要人为引入空隙时, 使用 `\quad` 和 `\qquad` 等命令。
2. 不允许有空行 (分段), 公式也无法自动换行或者用 `\\` 换行。排版多行公式需要用到各种环境。
3. 所有的字母被当做数学公式中的变量处理, 字母间距语文本模式不一致, 也无法生成单词之间的空格。如果想在数学公式中输入正体的文本, 简单情况下可以用 `\mathrm` 命令。或者用 `amsmath` 提供的 `\text` 命令。

示例:

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0$$

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}$$

数学家们对应该使用什么符号很挑剔: 上面的公式最好使用 “blackboard bold” 字体, 可以使用 `amssymb` 宏包里的 `\mathbb` 命令来完成。

示例:

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

3 构建数学公式块

在这一小节中大部分的命令都不需要引入 `amsmath` 宏包。

3.1 希腊字母

小写希腊字母通过 `\alpha`、`\beta`、`\gamma` 等输入, 大写希腊字母通过 `\Gamma`、`\Delta` 等输入。

示例:

$$\lambda, \xi, \pi, \theta, \mu, \Phi, \Omega, \Delta$$

3.2 指数，上标，下标

指数、上标和下标可以分别通过 \wedge 和 $_$ 指定。大多数的数学模式命令仅对它之后的那个字母起作用所以如果想对多个字母起作用，应该用 $\{...\}$ 括起来。

示例：

$$p_{ij}^3 \quad m_{\text{Knuth}} \quad \sum_{k=1}^3 k$$

$$a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}$$

3.3 根式

通过 $\sqrt[n]{}$ 输入平方根；通过 $\sqrt[n]{}$ 输入 n 次方根。根号的大小是由 \LaTeX 自动决定的，如果只需要一个符号标记，可以用 $\sqrt[n]{}$ 命令。

示例：

$$\sqrt{x} \Leftrightarrow x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt{[x^2 + y^2]}$$

3.4 点

示例：

$$\Psi = v_1 \cdot v_2 \cdot \dots \quad n! = 1 \cdot 2 \cdot \dots (n-1) \cdot n \quad \vdots \quad \ddots$$

3.5 横线

利用 $\overline{}$ 和 $\underline{}$ 命令来产生上划线和下划线。

示例：

$$0.\overline{3} = \underline{\underline{1/3}}$$

3.6 水平花括号

利用 $\overbrace{}$ 和 $\underbrace{}$ 命令来产生水平的上、下花括号。

示例：

$$\overbrace{a+b+c}^6 \cdot \overbrace{d+e+f}^7 = 42$$

$$\underbrace{a+b+c+d+e+f}_{\text{meaning of life}}$$

3.7 数学重音符号

示例：

$$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 2$$

$$\hat{X}Y \quad \widehat{XY} \quad \bar{x}_0 \quad \bar{\bar{x}}_0$$

$$\tilde{a} \quad \text{\textit{widetilde{a}}}$$

注意：上面的示例中， \backslash 命令后跟了可选参数 $[5pt]$ 来增加额外的行距。

3.8 向量

在变量上加箭头来表示向量，可以通过`\vec` 命令完成。

示例：

$$\vec{a} \quad \vec{AB} \quad \overrightarrow{AB}$$

3.9 函数名

在排版数学公式时，通常函数名不像变量那样用斜体，所以要用命令来进行区别。下面是 \LaTeX 中常用的函数命令：

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>
<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>	<code>\min</code>	<code>\Pr</code>
<code>\sec</code>	<code>\sin</code>				

对于上面没有列出的一些函数，可以在导言区用`\DeclareMathOperator` 或其加星版本的命令来定义。加星和不加星版本主要是在显示为行内公式还是行间公式上有区别。

示例：

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

$$3 \operatorname{argh} = 2 \operatorname{Nut}_{x=1}$$

3.10 取模函数

取模函数有两种命令：`\bmod` 和`\pmod`。

示例：

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

3.11 取模函数

可以通过`\frac{...}{...}` 命令来排版分式。在行内公式中，分式会收缩以和所在行保持一致。可以通过`\tfrac` 命令在行间公式强制显示为行内模式，同样地，也可以通过`\dfrac` 命令在行内公式中将分式强制显示为行间模式。一般，在行间公式中更经常使用斜杠的形式来表示比较短的分式，如 $1/2$ 。

示例：

In display style:

$$3/8 \quad \frac{3}{8} \quad \frac{3}{8}$$

In text style: $1\frac{1}{2}$ hours $1\frac{1}{2}$ hours

使用`\partial` 命令来输入偏导符号。

示例：

$$\sqrt{\frac{x^2}{k+1}} \quad x^{\frac{2}{k+1}} \quad \frac{\partial^2 f}{\partial x^2}$$

3.12 二项式系数和符号堆叠

可以通过 `amsmath` 宏包里的 `\binom` 命令来排版二项式系数

示例:

Pascal's rule is

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

对于一些二元关系, 有时候可能需要对符号进行堆叠, 这时候使用 `\stackrel{\#1}{\#2}` 命令。其中 `\#1` 会变成上标大小, `\#2` 会保持原来的大小和位置。

示例:

$$f_n(x) \stackrel{*}{\approx} 1$$

3.13 积分运算符、求和运算符、乘积运算符

分别用 `\int`、`\sum`、`\prod` 命令来输入积分运算符、求和运算符和乘积运算符。

示例:

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

在更复杂的表达式中, 可以通过 `amsmath` 宏包中的 `\substack` 命令来更好地控制索引的位置。

示例:

$$\sum_{\substack{0 \leq i \leq n \\ j \subseteq i}} P(i, j) = Q(i, j)$$

\LaTeX 可以输入各种括号和分隔符。圆括号和方括号可以通过各自的键位输入, 花括号需要使用 `\{` 命令, 其他的分隔符都要用特殊的命令 (如 `\updownarrow`)。

示例:

$$a, b, c \neq \{a, b, c\}$$

把 `\left` 命令放在左分隔符前面, 把 `\right` 放在右分隔符前面, 这样 \LaTeX 就会自动地调整分隔符大小。如果不要右半个分隔符, 使用不可见的 `\right.` 命令。

示例:

$$1 + \left(\frac{1}{1-x^2} \right)^3 \quad \dagger -$$

在有些情况下, 需要通过 `\big`、`\Big`、`\bigg`、`\Bigg` 来手动指定分隔符的大小。

示例:

$$\left((x+1)(x-1) \right)^2$$



4 长公式折行

有时候一个等式或方程太长了，有必要使其折行。但是折行会降低等式的可读性，为了提高可读性，在折行时可遵循以下规则：

1. 应该在等号或者某个操作符前折行
2. 优先在等号前而不是操作符前折行
3. 优先在加号、减号而不是乘号前折行
4. 避免其他的折行方式

实现这种折行的最简单方式就是使用 `amsmath` 宏包中的 `multline` 环境。

示例：

$$a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q \\ = r + s + t + u + v + w + x + y + z \quad (4)$$

与之相比，`equation` 环境可能会引入任意的一个或多个折行。而使用 `multline` 环境时，仅在需要折行的地方使用 `\\` 命令。和 `equation*` 类似，`multline*` 不对公式进行编号。

示例：

$$a = b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z + aa + bb + cc + dd + ee + ff \quad (5)$$

$$a = b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q \\ + r + s + t + u + v + w + x + y + z + aa + bb + cc + dd + ee + ff \quad (6)$$

5 多行公式

在遇到一些公式中有连等的情况时，我们希望在竖直方向上等号能够对齐。为了应对这个问题，看一些常用的方法以及他们的不足之处。

5.1 传统命令存在的问题

可以使用 `align` 环境来实现等号的对齐。

示例：

$$a = b + c \quad (7)$$

$$= d + e \quad (8)$$

但是在某一行公式过长时，使用 `align` 环境的输出会出现问题。如下面的示例，`+ v` 应该和 `d` 对齐，而不是和上面的等号对齐。可以通过在它前面加一些空格（`\hspace{...}`）来实现我们想要的效果，但是这不够准确。

示例：

$$a = b + c \quad (9)$$

$$= d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u \\ + v + w + x + y + z + aa + bb + cc \quad (10)$$

$$= dd + ee + ff + gg + hh + ii \quad (11)$$

`eqnarray` 环境提供了一种更好的解决方案。

示例：

$$a = b + c \quad (12)$$

$$= d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u \\ + v + w + x + y + z + aa + bb + cc \quad (13)$$

$$= dd + ee + ff + gg + hh + ii \quad (14)$$

但使用 `eqnarray` 环境仍不是最优的解决方案。等号两边的太大，而且和 `multline` 环境 `equation` 环境中的大小不一样。而且有些时候即使左边有足够空间，右边还是会和公式编号重叠。

示例：

$$a = a = a \quad (15)$$

$$a = b + c \quad (16)$$

$$= d + e + f + g + h^2 + i^2 + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \quad (17)$$

`eqnarray` 环境提供了 `\left` 命令，可以再公式的等号左边部分过长时使用。但是这样并不是最优的解决方案，而且等号右边部分过短时公式可能不会居中。

示例：

$$a + b + c + d + e + f + g + h \\ = i + j + k + l + m \quad (18)$$

$$= n + o + p + q + r + s \quad (19)$$

$$\begin{aligned} a + b + c + d + e + f + g + h \\ = i + j \end{aligned} \quad (20)$$

5.2 IEEEeqnarray 环境

为了使用 IEEEeqnarray 环境，需要在导言区使用一下命令引入 IEEEtrantools 宏包：

`\usepackage[retainorgcmds]{IEEEtrantools}`。

IEEEeqnarray 的一个优点是可以指定公式排列的列数。通常情况下，指定参数为 `{rCl}`，表示总共有三列，第一列右对齐，中间列居中（和小写字母 *c* 相比，大写字母 *C* 还会使其左右两边留有空隙），第三列左对齐。

示例：

$$a = b + c \quad (21)$$

$$\begin{aligned} &= d + e + f + g + h + i + j + k \\ &\quad + l + m + n + o \end{aligned} \quad (22)$$

$$= p + q + r + s \quad (23)$$

可以指定任意多列：`{c}` 表示只有居中的一列；`{rCl}` 会增加左对齐的第四列，一般用于添加注释。

5.3 IEEEeqnarray 环境的一般用法

如果某行公式会和公式编号重叠，可以在对应公式行后使用 `\IEEEeqnarraynumspace` 命令来解决。

示例：

$$\begin{aligned} a &= b + c \\ &= d + e + f + g + h^2 + i^2 + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \end{aligned} \quad (24)$$

$$\begin{aligned} a &= b + c \\ &= d + e + f + g + h^2 + i^2 + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \end{aligned} \quad (26)$$

如果等号左边部分过长，IEEEeqnarray 提供了 `\IEEEeqnarraymulticol` 命令来取代有缺陷的 `\lefteqn` 命令。

示例：

$$\begin{aligned} a + b + c + d + e + f + g + h \\ = i + j \end{aligned} \quad (28)$$

$$= k + l + m \quad (29)$$

$$\begin{aligned} a + b + c + d + e + f + g + h \\ = i + j \end{aligned} \tag{30}$$

$$= k + l + m \tag{31}$$

`\IEEEeqnarraymulticol` 的用法和 `tabular` 环境中 `\multicolumns` 命令的用法是相同的。第一个参数 `{3}` 指定三列合并为一列，`{1}` 参数指定其左对齐。可以通过 `\quad` 等命令可以方便地控制等号的缩进。

如果公式折成了两行或多行，`LaTeX` 会把首个 `+` 或者 `-` 作为一个标识符，而不是运算符。因此需要在运算符和操作数之间插入额外的空格。

示例：

$$a = b + c \tag{32}$$

$$\begin{aligned} &= d + e + f + g + h + i + j + k \\ &\quad + l + m + n + o \end{aligned} \tag{33}$$

$$= p + q + r + s \tag{34}$$

$$a = b + c \tag{35}$$

$$\begin{aligned} &= d + e + f + g + h + i + j + k \\ &\quad + l + m + n + o \end{aligned} \tag{36}$$

$$= p + q + r + s \tag{37}$$

`IEEEeqnarray` 环境也有加星版本，会省略所有的公式编号，但可以使用 `\IEEEyesnumber` 和 `\IEEEyessubnum` 命令来显示某行公式的编号或子编号。

示例：

$$\begin{aligned} a &= b + c \\ &= d + e \\ &= f + g \end{aligned} \tag{38}$$

$$a = b + c \tag{38a}$$

$$\begin{aligned} &= d + e \\ &= f + g \end{aligned} \tag{38b}$$

6 数组和矩阵

可以通过 `array` 环境来排版数组，它的工作方式和 `tabular` 环境类似。

示例:

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots \\ x_3 & x_4 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

可以通过 `array` 环境来排版矩阵, 但是 `amsmath` 提供了一系列的 `matrix` 环境, 可以更好地排版矩阵。共有六种不同的分隔符: `matrix` (无)、`pmatrix` (`()`)、`bmatrix` (`[]`)、`Bmatrix` (`{}`)、`vmatrix` (`|`) 和 `Vmatrix` (`||`)。和 `array` 环境不同, `matrix` 环境不需要指定列数, 最大列数为 10。

示例:

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}$$

7 数学模式中的空格

如果 \LaTeX 数学公式中的空格不能满足要求, 可以通过插入一些特殊的命令进行调整。`\,` 命令对应 $\frac{3}{18}$ 个 quad; `\:` 命令对应 $\frac{4}{18}$ 个 quad; `\;` 命令对应 $\frac{5}{18}$ 个 quad; 转移空格字符 `_` 会产生一个和单词间距大小相当的空格; `\quad` 会产生一个当前字体下 ‘M’ 字母宽度的空格, `\qquad` 相当于两个 `\quad`; `\!` 命令会产生一个 $-\frac{3}{18}$ 个 quad。

示例:

$$\int_1^2 \ln x dx \quad \int_1^2 \ln x \, dx$$

注意微分运算中的字母 ‘d’ 一般用罗马字体。下面的李子辉定义一个新命令 `\ud` (`upright d`), 可以实现相同的效果。

示例:

$$\int \int f(x)g(y) \, dx \, dy$$

$$\iint f(x)g(y) \, dx \, dy$$

$$\int\!\!\int f(x)g(y) \, dx \, dy$$

7.1 幽灵

`\phantom` 命令可以为其中的内容保留位置，但这些内容又不会在最终的输出中出现。

示例：

$${}^{14}_6\text{C} \quad \text{versus} \quad {}^{14}_6\text{C}$$

`mhchem` 宏包可以更方便地输入同位素和化学方程式。

8 折腾数学字体

示例：

\Re \mathcal{R} *quad* \Re \mathbb{R}

上面的后两种字体命令需要用到 `amssymb` 或 `amsfonts` 宏包。

有时候需要给 \LaTeX 制定合适的字体大小。在数学模式中，可以通过以下命令进行设定：

`\displaystyle(123)`, `\textstyle(123)`, `\scriptstyle(123)` and `\scriptscriptstyle(123)`

如果 *sum* 符号放在分式中，它会默认排版成文本模式，所有可以根据需要用 `\displaystyle` 进行指定。

示例：

$$p = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[\sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$$

8.1 粗体符号

在 \LaTeX 中打印粗体符号有一些困难。`\mathbf` 命令可以打印粗体的英文文本，但是打印的是罗马字体，而数学公式中的字母大都是斜体的，而且该命令对希腊字母无效。`\boldmath` 命令可以打印粗体的英文字母和希腊字母，但是他不能用在数学公式中，只能包裹在数学公式外面。

`ambsy` 宏包（包含于 `amsmath`）和 `bm` 宏包（包含于 `tools`）中都有一个 `blodsymbols` 命令，可以比较方便地打印粗体符号。

示例：

μ, M μ, \mathbf{M} $\boldsymbol{\mu}, M$ $\boldsymbol{\mu}, \mathbf{M}$

9 定理，引理.....

在撰写数学文档的时候，可能需要排版、定理、公理等。可以通过下面的命令来完成：

`\newtheorem{name}[counter]{text}[section]`

其中，`name` 参数使我们定义定理的名称，作为一个环境来使用；`text` 参数是定理真正的名字，会显示在最终的文档中。方括号中的参数是可选的，定理的序号由两个可选参数之一决定，它们不能同时使用。`section` 为章节名称，使定理序号成为章节的下一级序号。`counter` 可以为用 `\newcounter` 自定义的计数器名称，则使用一个默认的计数器。

例如, 我们用`\newtheorem{mythm}My Theorem[section]` 命令定义了一个 `mythem` 环境, 然后就可以用它来排版定理。定理带有一个可选参数, 可以用于注明定理的名称。

示例:

My Theorem 9.1. *The light speed in vaccum is **299,792,458 m/s***

My Theorem 9.2 (Energy). *The relationship of engergy,momentum and mass is*

$$E^2 = m_0^2 c^4 + p^2 c^2$$

where c is the light speed described in theorem 9.1.

`amsthm` 宏包提供了`\theoremstyle{style}` 命令, 可以定义不同的 `theorem` 类型, 包括 `definition`、`plain`、`remark` 三种。

```
\theoremstyle{definition} \newtheorem{law}{Law}
\theoremstyle{plain}      \newtheorem{jury}[law]{Jury}
\theoremstyle{remark}     \newtheorem*{marg}{Margaret}
```

示例:

Law 1. Don't hide in the witness box

Jury 2 (The Twelve). *It could be you!So beware and see law 1.*

Jury 3. *You will disregard the last statement.*

Margaret. No , No , No ,

Margaret. Denis!

在上面的示例中,“Jury”定理和“Law”定理用了相同的 `counter` 参数, 所以“Jury”的编号是接着“Law”的。

示例:

Murphy 9.1. If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.

示例中的“Murphy”定理由于定义时指定了 `section` 参数, 所以它的编号和当前章节有关。`section` 参数也可以换成 `chapter` 或 `subsection`。

9.1 证明和证毕符号

`amsthm` 宏包也提供了 `proof` 环境。

示例:

证明. Trival, use

$$E = mc^2.$$

□

如果行末是一个不带编号的公式, 证毕符号会另起一行, 这时候可以使用`\qedhere` 命令将证毕符号放在公式末尾。`\qedhere` 命令对 `IEEEeqnarray` 无效。这是由于 `IEEEeqnarray` 的内部结构导致的。为了保证公式可以水平居中, `IEEEeqnarray` 在公式块两边各加了一列。这两列只包含一个可以拉伸的空格, 是不可见的, 因此`\qedhere` 命令无法放在可拉伸空格的外面。

示例:

证明. This is a proof that end with an equation array:

$$\begin{aligned} a &= b + c \\ &= d + e. \quad \square \end{aligned}$$

`{+rCl+x*}` 中的 `+` 表示可拉伸的空格, 在公式左右各一列。现在又在右边可拉伸的空格外加了一个空白列 `x`, 这一列只有最后一行的`\qedhere` 命令需要。最后又指定一个 `*`, 表示没有空格, 放置 `IEEEeqnarray` 在最右边添加额外的 `+` 空格。

示例:

证明. This is a proof that ends with a numbered equation:

$$a = b + c. \tag{39}$$

□

证明. This is a proof that ends with a numbered equation:

$$a = b + c. \tag{40}$$

□

证明. This is a proof that ends with a numbered equation:

$$a = b + c \tag{41}$$

$$= d + e. \tag{42}$$

□

证明. This is a proof that ends with a numbered equation:

$$a = b + c \tag{43}$$

$$= d + e. \tag{44}$$

□