

TeX

for the
Impatient

Paul W. Abrahams

with

Karl Berry

Kathryn A. Hargreaves



TEX

2025 年 10 月 6 日

‘ \TeX ’ is a trademark of the American Mathematical Society.

‘ \METAFONT ’ is a trademark of Addison-Wesley Publishing Company.

This book, \TeX , contains both tutorial and reference information on all features of both plain and primitive \TeX .

Copyright © 2003, 2013 Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover texts, and with no Back-Cover texts. A copy of the license is included in the chapter entitled “GNU Free Documentation License”.

Under the terms of the GFDL, anyone is allowed to modify and redistribute this material, and it is our hope that others will find it useful to do so. That includes translations, either to other natural languages, or to other computer source or output formats.

In our interpretation of the GFDL, you may also extract text from this book for use in a new document, as long as the new document is also under the GFDL, and as long as proper credit is given (as provided for in the license).

Jodi.

—P.W.A.

.

—K.A.H.

Dan.

—K.B.

Donald Knuth T_EX T_EX METAFONT T_EX METAFONT T_EX
T_EX
T_EX T_EX T_EX T_EX
T_EX T_EX T_EX T_EX T_EX
T_EX T_EX T_EX The T_EXbook ‘resources’ T_EX The T_EXbook
T_EX T_EX T_EX
T_EX T_EX

T_EX T_EX The T_EXbook The T_EXbook T_EX
1989 Knuth T_EX 8 T_EX ‘newtex’).

你可能正在使用一个专门的 T_EX 封装形式，比如 L^AT_EX 或者 $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX（见第 ‘resources’ 页）。虽然这些封装形式是完整的，你仍可能为更好地控制 T_EX 而去使用 T_EX 本身独有的一些功能，你可以使用这本书来学习你想要知道的这些功能，而把其它你不感兴趣的东西扔在一边。

在准备本书写作时，我们作者中的两位 (K.A.H. 和 K.B.) 获得波士顿麻省大学的鼎力帮助，尤其是 Rick Martin 在此期间，保证计算机的正常运行，以及 Robert A. Morris 和 Betty O'Neil 能让我们使用这些计算机设备。Paul English of Interleaf 帮助我们做出了封面设计的校样。

我们需要感谢本书的这些审稿人：马里兰大学的 Richard Furuta，Arbortext 公司的 John Gourlay，电子设备公司的 Jill Carter Knuth 和 Richard Rubinstein。我们把他们对我们原稿的观点和批评铭记在心，本书从他们的洞见中获益良多。

我们感谢我们的编辑，Addison-Wesley 出版公司的 Peter Gordon。出版本书其实是他的主意，在本书的写作过程中，他也鼓励我们并且提供有价值的建议。我们感谢他在 Addison-Wesley 的助理，Helen Goldstein，因为她在很多方面帮助我们。我们感谢 Addison-Wesley 的

Loren Stevens 在指导本书出版过程中的发挥的激情和技艺。如果没有我们的技术编辑 Janice Byer, 很多小却恼人的错误可能还留在这本书中。我们感谢她在区分需要和不需要改动的地方的敏感和品位。最后, 我们需要感谢 Prometheus 有限公司的 Jim Byrnes, 他介绍我们互相认识, 使我们能够有机会进行合作。

Deerfield, Massachusetts
Manomet, Massachusetts

P. W. A.
 K. A. H., K. B.

这本书本来是 Addison-Wesley 在 1990 年出版的。在 2003 年售空, Addison-Wesley 慷慨地把所有的权力归还给我们作者。没有社区, 也不会有本书的存在, 所以作为我们对社区的贡献, 我们打算在 GNU 的自由文档许可协议下把这本书开源, 更多的关于许可协议的信息, 见本书的版权页。

原本本书包括了一些插图, 不过在自由版本中, 我们没有把插图包括进来。此外我们还改变了一些字体, 只使用可自由获取的字体。我们把剪裁线和长条校样保留在页面上以方便识别。本书使用老版本的 Eplain 宏包编写, 更多信息见 `eplain.tex` 文件。

除了修正报告给我们的错误, 以及可能加入原书的插图以外, 我们打算修改或者增加本书的内容。

我们把本书发布在 `ftp://tug.org/tex/impatient` 上。你可以通过电子邮件 `impatient@tug.org` 来和我们取得联系。

本书的翻译由 CTeX 论坛上多人合作完成。在 2008 年, yulewang firingme、LiYanrui、Neals、sde、snoopyzhao zpxing 2009

2014 zoho

<https://github.com/CTeX-org/tex-impatient-cn>

<http://bbs.ctex.org/forum.php?mod=viewthread&tid=78586>

qingkuan

The T_EXbook The T_EXbook

|

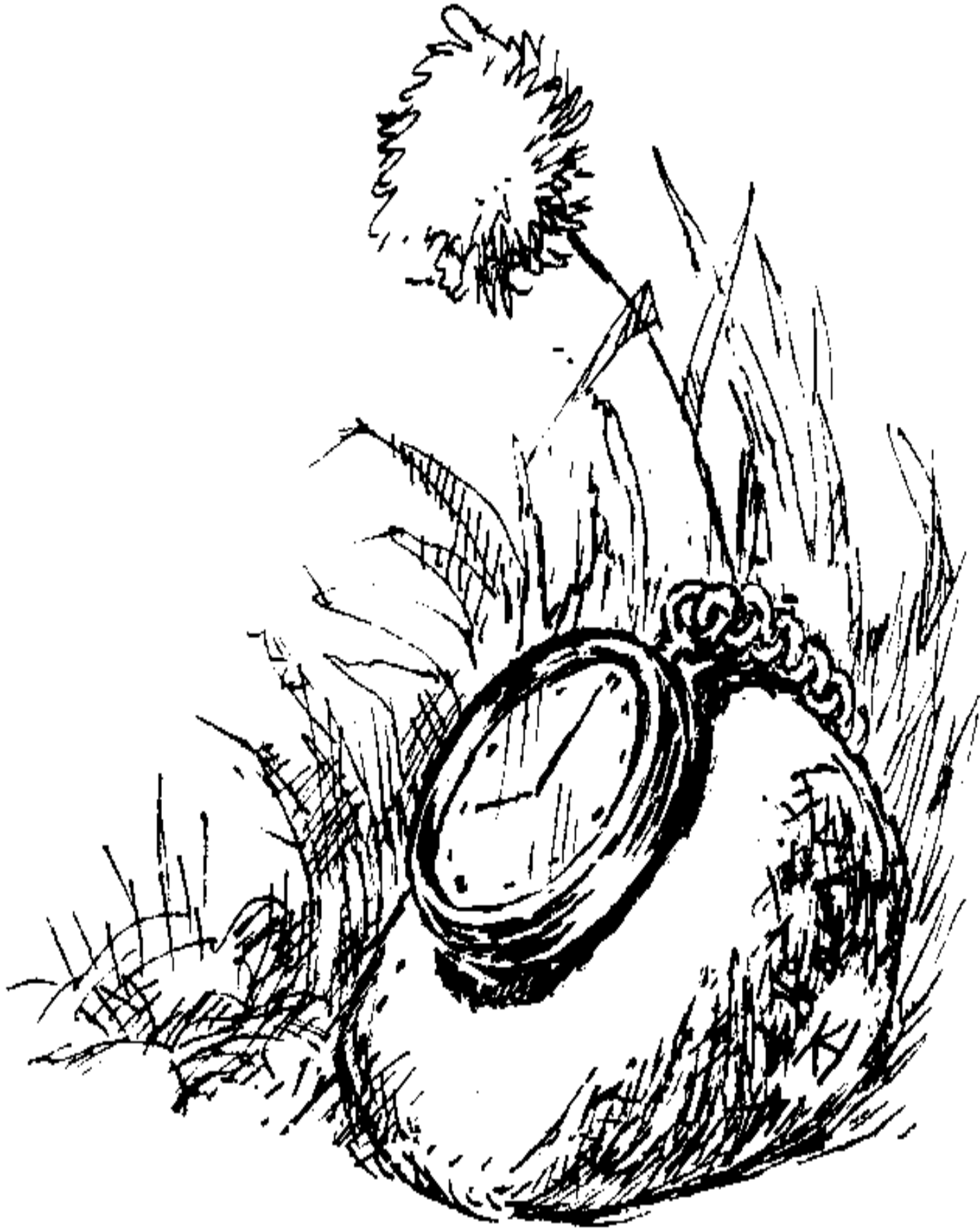
|

T_EX

- ‘usebook’–‘usingt_Ex’ 章。
- 查看第 ‘examples’ 章中与你要做的事情相似的例子。查找第 ‘capsule’ 章，“命令速查表”中相关的命令。利用页码索引找到这些命令及类似命令的更完整的描述。
- 查找第 ‘concepts’ 章，“概念”中不熟悉的词语，使用书背内页的概念列表快速找到这些词语的解释。
- 反复尝试和探索。

T_EX T_EX

- ‘capsule’
-
- ‘concepts’



1

T_EX

‘usebook’ ‘examples’ 章，这些地方会告诉你如何上手 T_EX。如果你已经使用过 T_EX，这些地方仍会值得一看，至少你可以知道这些章节介绍了些什么内容。其余部分，主要是‘concepts’--‘tips’章，可以有选择地翻阅。当然，如果你是一个很喜爱阅读参考手册的人，你会发现从头到尾地把这本书阅读一遍完全可行的，当然你会为此在开头学习时绕些弯路。

在第‘usintex’章，‘使用 T_EX’，我们解释如何把一个 T_EX 输入文件转为一个 T_EX 文档。我们还讲述了准备输入文件的一些习惯，简要地解释了 T_EX 是如何工作的，告诉你可以获取的其它资源信息。阅读了这个章节，会有助于你理解下一个章节的例子。

第‘examples’章，‘例子’，包括了一系列的例子来描述 T_EX 的功用。每个例子包括了一页的输出内容，也包括了输出该页内容所输入的源代码。这些例子会指导你且帮助你找到你所需要的更详细的资料。通过观察输入使用了什么命令，你会知道在哪里能够找到输出效果所使用方法的更详细的信息。这些例子也可以作为简单的文档的模版，当然我们必需提醒你，因为我们尝试把很多的 T_EX 命令放在这几页纸中，这些例子不可能描述好的或者完整的文档设计。

当你看这些命令的解释时，你会遇到很多不熟悉的技术术语。在第‘concepts’章，‘概念’中，我们定义并解释了这些术语。我们同时还讨论了本书其它地方没有涉及到的话题。本书书背内页列出了所有的概念名称以及它们所出现的位置。我们建议你用书背内页复印下来，放在旁边，这样你就能在遇到不熟悉的概念时马上查找到它们。

T_EX 的命令是它的主要词汇，本书最主要的部分就是用来解释这些命令。从第 ``paras'` 章到第 ``general'` 章，我们解释了这些命令。你可以在第 ``cmddesc'` 页找到命令描述的描述惯例。这些命令根据功能分门别类，而不是按字典顺序，所以如果你知道你想做什么，但不知那该用什么命令，你可以使用目录来引导你找到正确的命令组。我们给我们认为是常用且容易理解的命令标上了一个手形符号 (✎)。

第 ``capsule'` 章为“命令速查表”，这个专门的索引与第 ``paras'--`general'` 章的更完整描述相辅相成。在这一章中按字母顺序列出了各个 T_EX 命令及其简要解释，并给出该命令的详细描述所在的页码。此速查表可以让你快速知道某个命令的作用。

T_EX 是个复杂的程序，偶尔地会以诡异的方式工作。在第 ``tips'` 章，“建议和技巧”中，对一系列你时不时会遇到的特定的问题，我们提供了一些建议。而若你被 T_EX 的错误信息难住了，第 ``errors'` 章，“理解错误信息”将给你提供帮助。

页边的灰色标签可以帮你快速定位到本书各个部分。它们将本书划分为下面几个主要部分：

- 1) 总体的说明和例子
- 2) 概念
- 3) 命令描述（五个短标签）
- 4) 建议，错误信息和 `eplain.tex` 宏集
- 5) 命令速查表
- 6) 索引

在很多地方我们都给出了到 *The T_EXbook* 的页码索引（要引用该书可以见第 ``resources'` 页）。这些索引适用于 *The T_EXbook* 第十七次印刷版。对于其它版本，有的索引可能相差一两页。

在任何涉及到为计算机准备输入的书籍中，都得明确区分哪些字符是需要逐字输入的，而哪些字符属于解释性文本。我们用计算机现代打字机字体显示文字输入，比如 `literal input`，以及 T_EX T_EX ‘`like this`’。然而，在表示单个字符时，我们偶尔也会用括号将它围起来，比如 (‘)（其原因你容易明白）。

通常我们按照你输入空格的方式显示空格，以免让你看花眼。但为了强调某些空格，我们用‘`␣`’字符表示它。这个字符很自然地被称为。

第 ``paras'--`general'` 章描述了几乎每个 `TEX` `plain TEX` `TEX` `plain TEX` ‘`plainTEX`’ `plain TEX` (*The T_EXbook* B) 定义中用到的局部命令。这些命令分为如下几部分：

- 第 ``paras'` 章，```组段命令`"，介绍字符、单词、文本行和整个段落。
- 第 ``pages'` 章，```组页命令`"，介绍页面及其组成部分，以及输出例行程序。
- 第 ``hvmodes'` 章，```水平和竖直模式命令`"，介绍水平模式（段落和水平盒子）和竖直模式（页面和竖直盒子）中对应或等同的命令。这些命令提供了盒子、间隔、标线、指引线和对齐。
- 第 ``math'` 章，```数学公式命令`"，介绍构造数学公式的命令。
- 第 ``general'` 章，```一般操作命令`"，介绍 `TEX` 的编程功能以及不适合放在其他部分的命令。

你应该将这些分类视为提示性的而不是严格的，因为把命令都归入这些（或者其他）类别并不完全合适。

各章的命令描述按照其功能组织。当几个命令密切相关时，它们合起来介绍；否则，每个命令分别解释。命令的描述中包含一个或多个例子，可能还有例子的输出（对有些例子这并不可行）。若某节内容涉及到功能上相关的其他命令时，务必看看该节末尾的“参见”项，那里给出了其它地方介绍到的相关命令所在的页码。

有些命令与特定的概念密切相关。例如，`\halign` 和 `\valign` 命令与“对齐”的概念相关，`\def` 命令与“宏”相关，而 `\hbox` 和 `\vbox` 命令与“盒子”相关。对此种情形，在命令这里我们通常只给出粗略介绍，而将对基本原理的解释放在相关概念中。

命令相关的例子在排版时设定了段落缩进 `\parindent` 为零，因而其中段落一般是不缩进的；这使得例子更加容易阅读。对有些例子段落缩进是必不可少的，我们显式设置缩进为非零值。

在一个或一组命令前面的手形符号，表示我们认为这个或这组命令是常用且容易理解的。

某些命令需要特定类型的参量 (第 `\arg1` 页)。命令的参量提供 TeX 执行该命令时所需的额外信息。每个参量都放在尖括号中, 并用意大利体标明它的类型:

<code>\argument</code>	单个记号或者放在花括号中的一些文本
<code>\charcode</code>	字符码, 即在0到255之间的整数
<code>\dimen</code>	尺寸, 即长度
<code>\glue</code>	粘连 (有可选的伸展值和收缩值)
<code>\number</code>	可带符号的整型值 (整数)
<code>\register</code>	寄存器编号, 在0到255之间

所有这些术语会在第 `\concepts` 章中进行详述。除了这些以外, 我们会使用一些不言自明的或者在后文进行讲述的术语, 比如 `\token list`。有些命令有自己独特的格式, 比如需要括弧或者特定的字词。它们会使用与命令标题相同的黑体标出。

有些命令是参数 (第 `\introparms` 页) 或表格项。这会在命令的列表中指示出来。你可以将这个参数作为一个命令的参量, 也可以给它赋值。表格项也是如此。我们使用“参数”这个术语来指代这些条目, 比如本是寄存器但可当参数使用的 `\pageno`。



2 | T_EX



```
TEX TEX TEX TEX TEX TEX. TEX
TEX TEX. TEX
TEX \input (第`input' 页) \input TEX TEX TEX TEX
plain TEX (第`plainTEX' 页)
TEX .dvi “dvi” “device independent (设备无关)”. .dvi
.dvi . (浏览器是一个程序，它能把和打印出来的结果差不多的排
版内容显示在屏幕上。) TEX
TEX TEX () () .tfm, .pk, .pxl .gf TEX cmr10.tfm
cmr10 (10-点的计算机现代字体).
TEX
```

■ T_EX

```
‘run tex’ ‘tex’ (查看你的本地信息) TEX screed.tex. TEX
This is TeX, Version 3.0 (preloaded format=plain 90.4.23)
**
```


“preloaded format” (已预先载入的格式) \TeX plain \TeX ‘screed’ \TeX
(如果你不使用终端)

```
(screed.tex [1] [2] [3] )
Output written on screed.dvi (3 pages, 400 bytes).
Transcript written on screed.log.
```

```
 $\text{\TeX}$  .dvi  $\text{\TeX}$  ‘.tex’.  $\text{\TeX}$ 
tex screed
```

\TeX ‘**’ ‘screed’ ‘\relax’。 \TeX 将在你输入的各行前面给出 ‘*’ 提示，并且逐行解释你的输入。输入类似 ‘\bye’ 这样的命令可以结束输入。在试验 \TeX 时直接输入有时更方便。

当你的输入文件包含其他嵌入的输入文件时，显示的信息中表明了 \TeX 何时开始和结束处理每个嵌入文件。当 \TeX 开始处理一个文件时它显示一个左圆括号和该文件名，而结束时显示一个对应的右圆括号。如果在所显示的输出中有任何错误信息，你就能从最接近的未配对左圆括号中确定是哪个文件引起的。

要得到如何运行 \TeX *The \TeX book* 6。

\TeX ‘examples’

■

\TeX \TeX “排版我”。比如，字母 ‘a’ 是作为命令将排版出 ‘a’。但有另一种命令——(control sequence)——将给 \TeX 提供更细致的指令。控制序列通常以反斜杠 (\) 开头，但必要时你也可以更改此约定。例如，下列输入：

```
She plunged a dagger (\dag) into the villain’s heart.
```

包含一个控制序列 \dag；它排版出下列的输出：

```
She plunged a dagger (†) into the villain's heart.
```

在这个例子中，除 \dag 和空格之外的每个字符都如同一个“排版我”命令。在第 ‘spaces’ 页中我们将更详细地解释空格的作用。

控制序列可分为 (control word) 和 (control symbol) 这两种类型：■

- 控制词由反斜杠后跟一个或多个字母组成，例如 ‘\dag’。其后用一个非字母字符结束该控制词。
- 控制符由反斜杠后跟一个单个的非字母字符组成，例如 ‘\\$’。该非字母字符可以是空格符，甚至是行结束符（这也是合乎规则的字符）。

控制词后的任何空格或行结束符将被忽略（但控制符后不会这样）。如果你真要在控制词后得到一个空格，可以在该控制词后键入一个控制空格（_）或者 ‘{ }’。因而这样输入：

```
The wonders of \TeX\_shall never cease!
```

或者这样输入：

```
The wonders of \TeX{ } shall never cease!
```

将得到同样的结果：

The wonders of T_EX shall never cease!

作为对比，下面的结果：

The wonders of T_EXshall never cease!

是你去掉上面输入中的 ‘_’ 或者 ‘{ }’ 后得到的。

不要将控制词和其后的文本连在一起---这样 T_EX \c garçon‘gar\c_con’, ■ 而不是 ‘gar\ccon’；若你输入后者，T_EX \ccon

‘\$13.56’ 你必须键入 ‘\\$13.56’，而不是 ‘_13.56’；后者将生成 ‘\$ 13.56’。然而，那些作为控制符的重音命令，是以忽略其后空格的方式定义的。例如，要得到法文单词 *déshabiller*，键入 ‘d\’eshabiller’，或者 ‘d_eshabiller’ 都可以。

每个控制序列同时也是一个命令，但反之未必。例如字母 ‘N’ 是一个命令，但它不是一个控制序列。在本书里面，当两者都适用时我们用“命令”而不用“控制序列”。在强调 T_EX 的语法层面时，我们用“控制序列”，因为此时用“命令”一般并不合适。

■

有些命令的运行依赖于其后的一个或多个 (argument)。例如，\vskip命■令告诉 T_EX 在页面中往上（或往下）跳过一定间距，它需要一个参量来指定所跳过间距的大小。要往下跳过两英寸，你需要键入 ‘\vskip 2in’，其中的2in就是\vskip的参量。

不同的命令要求不同类型的参量。很多命令的参量要求是长度值，比如上面例子中的`2in`。有些命令，特别是用宏定义的命令，其参量可以是单个字符或者放在花括号中的文本。而对其他有些命令，其参量只能放在花括号中，而不能是单个字符。在本书对各个命令的描述中，若一个命令需要参量，我们都说明了该命令所需参量的类型。在有些情形中，命令的参量所需的花括号定义了一个编组（见第 `'bracegroup'` 页）。



有些命令实际上是参数（parameter，见第 `''` 页）。你可以按下面两种方式使用这种参数：

- 1) 将该参数的值用作其他命令的参量。例如 `\vskip\parskip` 这个命令生成大小等于粘连参数`\parskip`（段落间距）的竖直间距。
- 2) 通过赋值修改该参数的值。例如 `\hbadness=200` 这个赋值使得`\hbadness`参数的值变为200。

有些命令，例如`\pageno`，虽然是寄存器但和参数的用法一样。因此我们同样使用“参数”这个术语来指代这些命令。

有些命令是表名。这些命令的用法和参数类似，只是它们需要额外的参量指定表格项。例如，`\catcode` 是类别码表格的名称（第 `''` 页）。因此，`\catcode'\~ = 13` 这个命令设置字符 `'~'` 的类别码为13。



在输入文件中你可以任意加上额外的空格。在几乎所有情形中 `TEX` 都将同一行的连续多个空格看成一个空格。例如，在句号后加上一个或两个空格是没有差别的。不论按照哪种写法，`TEX` `TEX` `TEX`

`TEX` `TEX`
`'unwantedspace'`

“粘连”（第 `''` 页）；粘连有其自然宽度（它希望的宽度），但也可以伸展或者收缩。`TEX` 排版要求右页边对齐（通常如此）的段落时，将调整各行的粘连的宽度以让各行正好在页边结束。（段落的最后一行是个例外，因为一般不要求它在右页边结束。）

要阻止输入中的空格在输出中变成行结束符，你可以用带子（`~`）。比如，你不会希望 `TEX` 在 `'Fig. 8'` 的 `'Fig.'` 和 `'8'` 之间断行。输入 `'Fig.~8'` 就可以禁止这样断行。



在 TEX TEX

```
% ===== Start of Section 'Hedgehog' =====
```

```
% TEX TEX
```

```
A fool with a spread%
sheet is still a fool.
```

A fool with a spreadsheet is still a fool.



‘.’, ‘?’ ‘!’ TEX TEX

```
A computer from IBM\null?
```

\null TEX ‘M’ 和问号结合在一起。反过来，如果某标点符号不为句末标点，你也能够取消该标点之后的额外间隔；这可以通过在其后键入控制空格达到，例如：

```
Proc.\_Royal Acad.\_of Twits
```

得到的正确结果为：

```
Proc. Royal Acad. of Twits
```

而去掉控制空格后的结果是：

```
Proc. Royal Acad. of Twits
```

有些人更喜欢不在句末标点后留下更多间隔。利用 `\frenchspacing` 命令（第 `'frenchspacing'` 页）你就可达成此效果。在参考文献中通常建议使用这个 `\frenchspacing` 命令。

对于单引号，用键盘上的左右单引号（‘和’）depth height plus 0.1em minus 0.1 em 就可以得到。对于左双引号或右双引号，需要用两个左单引号（‘‘）depth height plus 0.1em minus 0.1 em 或右单引号（’’）depth height plus 0.1em minus 0.1 em 得到，不能直接用键盘上的双引号（"）depth height。键盘上的双引号在多数字体中将得到一个右双引号，但两个右单引号是 TEX 中首选的写法。例如：

```
There is no 'q' in this sentence.
'Talk, child,' said the Unicorn.
She said, '\thinspace'Enough!', he said.'
```

这三行输入得到的结果为：

```
There is no 'q' in this sentence.
"Talk, child," said the Unicorn.
She said, "'Enough!', he said."
```

第三个输入行中的 `\thinspace` 避免单引号和双引号太过靠近。否则，你将看到三个并排的间隔相近的引号。

在 \TeX 中有三种横线号 (dash)：

- 短横线即连字号 (hyphen) `depth height plus 0.1em minus 0.1 em` 如这样 (-)。键入 ``-'` 可以得到。
- 中横线即连接号 (en-dash) 如这样 (--)。键入 ``--'` 可以得到。
- 长横线即破折号 (em-dash) 如这样 (---)。键入 ``---'` 可以得到。

一般地，你将用连字号表示像 “will-o'-the-wisp” (鬼火) 这样的复合词，用连接号表示像 “第 81--87 页” 这样的页码起止，而用破折号表示语气转折---像这样。



在 \TeX 中某些字符有特殊的含义，因而不能在普通文本中使用。这些特殊字符为：

```
$ # & % _ ^ ~ { } \
```

要在排版的文档中生成它们，你需要使用间接的方法。对前面五个字符，你需要键入：

```
\$ \# \& \% \_
```

而对另外五个字符，你需要键入更多：

```
\^{ } \~{ } $\{ $ }\$ \$\backslash$
```



包含在配对的左右花括号 ({ 和 }) `depth height plus 0.1em minus 0.1 em` 中的内容组成一个。编组内部的命令，其作用范围被限制在该编组内部。例如，`\bf` 命令让 \TeX 将某些内容用粗体显示。如果你将

`\bf` 放在你的输入中，而没有用任何方式取消它，`\bf` 之后的所有内容都将以粗体显示。如果将 `\bf` 包含在一个编组中，就可以将它的作用限制在此编组中。比如，如果你键入：

```
We have {\bf a few boldface words} in this sentence.
```

你将会得到如下的结果：

We have **a few boldface words** in this sentence.

利用编组，你可以限制 TEX 某个参数的取值的作用范围。这些参数的取值影响 TEX 对文档的排版。例如，`\parindent` 参数的值指定了段首缩进量，赋值 `\parindent = 15pt` 将设定该缩进量为 15 点。若将该赋值放在包含几个段落的编组开始处，你就可以仅仅改变这几个段落的段首缩进。若不将该赋值包含在编组中，对缩进量的修改将作用到文档的剩余部分（或者作用到下个对 `\parindent` 的赋值为止，若还有下个赋值的话）。

并非所有配对花括号都表示一个编组。特别地，若参量有花括号，则与该参量一起的花括号并不表示一个编组——它们仅用于确定该参量的界限。对于那些参量中确实需要花括号的命令，有些命令将花括号作为编组的定义，而其他命令用各自的特殊方式解释该参量。¹



数学公式可以出现在正文中 `() depth height`，或者出现在留出额外上下间距的单独一行中 `() depth height`。文内公式两边用单个美元符 `($) depth height plus 0.1em minus 0.1 em` 括起来，而陈列公式两边用双个美元符 `($ $) depth height plus 0.1em minus 0.1 em` 括起来。例如：

```
If $a<b$, then the relation $$e^a < e^b$$ holds.
```

此输入得到的结果是：

If $a < b$, then the relation

$$e^a < e^b$$

holds.

¹ TEX `\halign \valign`

第 ``math'` 章描述了数学公式中常用的命令。

T_EX

为了更有效地使用 T_EX，对 T_EX 着手将输入转换为输出的活动有所了解是有益的。你可以将 T_EX 想象为一种有“眼睛”、“嘴巴”、“食道”、“胃”以及“肠道”的生物体。该生物体的每个器官都以某种方式转换它的输入，并将转换的结果送到下个器官中。

眼睛将输入文件转换为一系列字符。嘴巴将这串字符转换为一系列，其中每个记号是单个字符或者一个控制序列。食道将该串记号展开为一系列记号。胃执行原始命令指定的操作，生成一些页面。最后，肠道将每个页面转换为 `.dvi` 文件所需的形式并发送给它。这些活动在第 ``concepts'` 章的“T_EX ” depth height (第 ``texpouxi'` 页) 中有更详细的描述。

实际的排版活动出现在胃中。这些命令指导 T_EX 用这样或那样的字体排版这个或那个字符，插入单词间空隔，结束一个段落，等等。从单个排版字符及其他简单排印元素开始，T_EX 将盒子一层层放入其他盒子中（见“ ” depth height，第 ``l'` 页），这一整套盒子构造了一个页面。每个排版字符占用一个盒子，整个页面也一样。盒子中不仅包含更小的盒子，也包含（第 ``l'` 页）和一些其他东西。粘连生成小盒子之间的间距。粘连的一个重要属性在于它可以伸展和收缩；因此通过伸缩盒子中的粘连，T_EX 可以让盒子变大或变小。

粗略说来，文本行是一个包含一系列字符盒子的盒子，而页面是一个包含一系列行盒子的盒子。一行的各单词之间以及页面的各行之间都有粘连。T_EX 伸缩各行中的粘连以让右页边对齐，伸缩各页中的粘连以让各页的下边距相等。其他类型的排印元素也可以出现在行内或页内，但我们不在这里讨论。

作为组装页面过程的一部分，T_EX 需要将段落分为多行，将多行组成页面。事实上，T_EX 的胃刚开始将段落看成一个很长的行。它插入以将段落分为长度合适的一些行，通过复杂的分析以找到最佳的断行点的集合（见“ ” depth height，第 ``l'` 页）。T_EX 的胃也执行类似但简单些的分析以将一些行组成页面。基本上，胃累加各行直到页面上放不下更多的行。然后它选择一个分页位置，将该分页位置之前的各行放在当前页面，将之后的各行留给下个页面（见“ ” depth height，第 ``l'` 页）。

在 T_EX 组装一个列表项（盒子，粘连等）中的某个东西时，它总处于六种（第 `'` 页）的其中一种模式中。它所组装的东西的类型定义了它所在的模式。首先有两种普通模式：普通水平模式用于组装段落（在它们分为多行之前），普通垂直模式用于组装页面。其次有两种限制模式：受限水平模式用于水平地添加元素形成水平盒子，内部垂直模式用于竖直地添加元素以形成垂直盒子（非页面盒子）。最后还有两种数学模式：文内数学模式用于组装段落内部的数学公式，陈列数学模式用于组装单独一行显示的数学公式（见“数学公式”，第 `'mathform'` 页）。

T_EX T_EX

在 1989 年，Knuth 对 T_EX 做了一次大幅修改，使得它能够处理非英语排版所需的字符集。这次修改还包括了一些额外的不干扰其它东西的小功能，这本书介绍“新 T_EX”。如果你仍使用旧版本的 T_EX（2.991 版或以前的版本），你可能会想知道哪些新 T_EX 的功能你是没法使用的。以下这些功能无法在旧版中使用：

- `\badness`（第 `'badness'` 页）
- `\emergencystretch`（第 `'emergencystretch'` 页）
- `\errorcontextlines`（第 `'errorcontextlines'` 页）
- `\holdinginserts`（第 `'holdinginserts'` 页）
- `\language`、`\setlanguage` 和 `\newlanguage`（第 `'language'` 和 `'@newlanguage'` 页）■
- `\lefthyphenmin` 和 `\righthyphenmin`（第 `'lefthyphenmin'` 页）
- `\noboundary`（第 `'noboundary'` 页）
- `\topglue`（第 `'topglue'` 页）
- 十六进制的数字表达方式 `^^xy`（第 `'hexchars'` 页）

我们建议你尽可能地使用新版本的 T_EX。

有很多资源可以帮助你使用 T_EX，其中 *The T_EXbook* 是最可靠的 T_EX 信息来源。

Knuth, Donald E., *The T_EXbook*. Reading, Mass.: Addison-Wesley, 1984.

请务必使用第十七次（1990 年 1 月）及以后的印刷版本。早先的印刷版本不包括新 \TeX 的许多功能。

\LaTeX 是一套为了简化 \TeX 的使用而设计的命令集，它在这本书里面有介绍²：

Lamport, Leslie, *The \LaTeX Document Preparation System*.
Reading, Mass.: Addison-Wesley, 1986.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 是一套美国数学学会定义的提交电子数学手稿的命令标准，它在这里有介绍：

Spivak, Michael D., *The Joy of \TeX* . Providence, R.I.: American Mathematical Society, 1986.

你可以加入 \TeX 用户组织 (TUG)，这个组织出版一本名为 *TUGBoat* 的通讯。TUG 是一个很好的信息来源渠道，它同时也提供包括 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 在内的很多 \TeX 宏包资源。它的地址是：

\TeX 用户组织
c/o American Mathematical Society
P.O. Box 9506
Providence, RI 02940
U.S.A.

最后，你可以获取在第 ‘explain’ 章介绍的用来排版本书的宏包 `eplain.tex`。■
它可以通过使用 `ftp` 匿名登录获取：

`labrea.stanford.edu` [36.8.0.47]
`ics.uci.edu` [128.195.1.1]
`june.cs.washington.edu` [128.95.1.4]

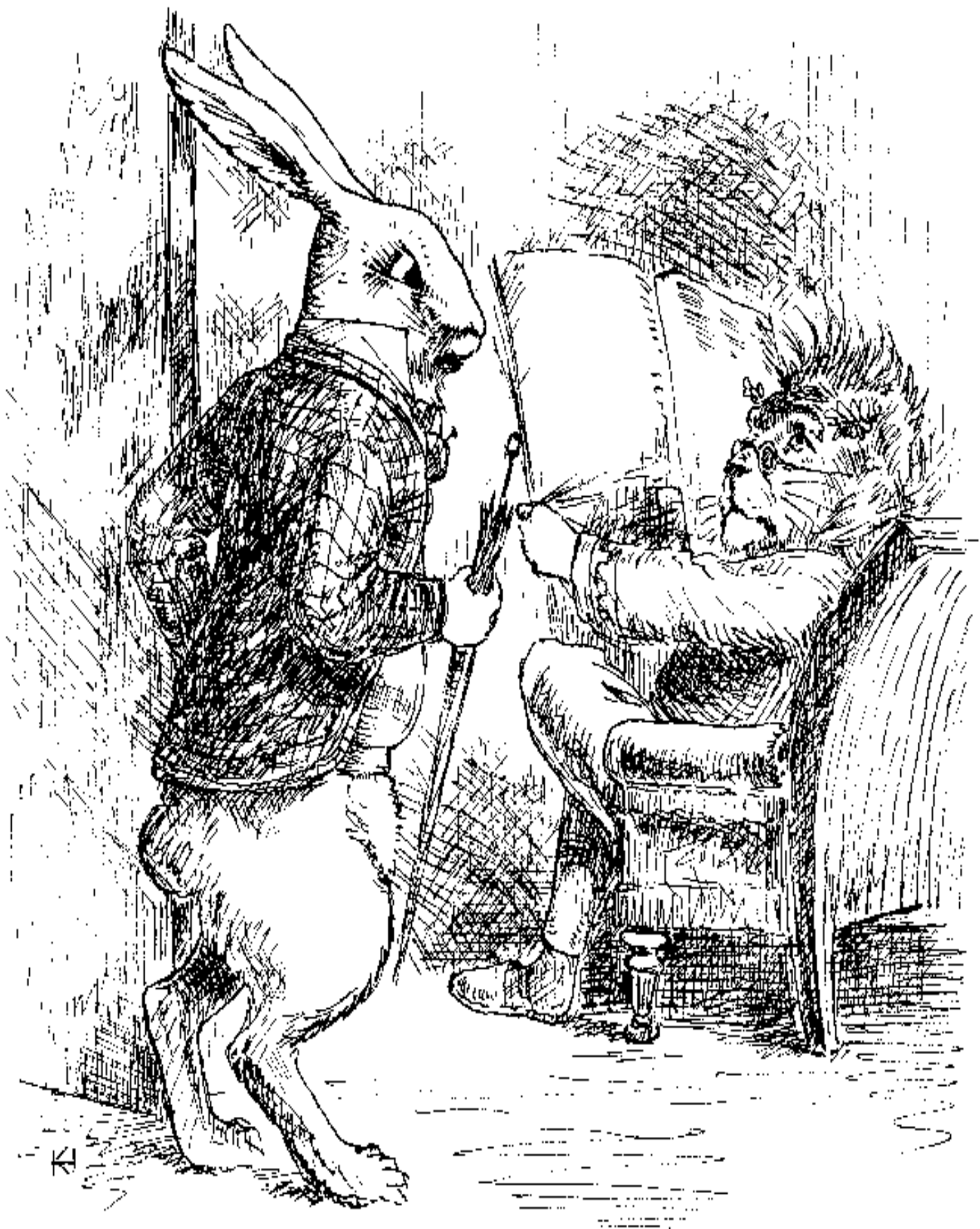
电子版本还包括了为 `Bib \TeX` 程序处理输入并排印其输出的宏——`Bib \TeX` 程序是斯坦福大学的 Oren Patashnik 编写的。如果你在这个宏包中发现了错误，或者想改进它，你可以给 Karl 发电子邮件。Karl 的电子邮箱地址是 `karl@cs.umb.edu`。

这个宏包还提供了 `5 1/4` 或 `3 1/2` 软盘，你可以向以下地址邮寄 10.00 美元购买：

Paul Abrahams
214 River Road
Deerfield, MA 01342
Email: `Abrahams%Wayne-MTS@um.cc.umich.edu`

² \LaTeX

这个地址在 1990 年 6 月是正确的；请注意在这之后它可能发生改变，尤其是电子邮件地址。



3



TEX TEX TEX TEX TEX 900

‘capsule’ TEX

8

\xmpheader “ \xmpheader \xmpheader \xmpheader Plain TEX

1:

T_EX T_EX † T_EX —

T_EX

T_EX T_EX DNA. “Fig. 8”)、姓名（如 “V. I. Lenin”）以及三个点...中，最好把单词绑定在一起，这样 T_EX 不会在它们的内部断行（三个点表示省略号）。

为得到正确的左右双引号，你需要用两个左右单“引号”。``在相邻的单引号和双引号之间，要插入一个`小间隔'`。连接号--可以这样写，而破折号---可以这样写。

† T_EX Tab

```

% TeX %
%
\font\xmplbx = cmbx10 scaled \magstephalf
\font\xmplbxti = cmbxti10 scaled \magstephalf
%
\leftline{\xmplbx 1:\quad\xmplbxti }
\vglue .5\baselineskip % skip an extra half line

\TeX\
\TeX\
%
\footnote \dag{\TeX\ Tab
{\it } 所见到的一样。}
\TeX\
---

\TeX\

\TeX\
\TeX\
DNA\null.
% 要输入这种带下划线的单词, 先定义一个控制字, 输入Lenin'小间隔'\thinspace'。

```


2:T_EX

6pt

3

“我们对这一段作了双边缩进的处理。在长段引用时，通常都会这么做。”

“如果你愿意的话，可以同时多个段落作如此处理。这是第二个轻微缩进的段落。”

```
\xmpheader 2/{缩进}%    'xmphead'
\noindent
\TeX\
```

```
{\parindent = 0pt \parskip = 6pt
%
```

```
6pt
```

```
\par % The paragraph must be ended within the group.
}% The right brace ends the group containing unindented text.
```

```
3
```

```
\smallskip % Provide a little extra space here.
% Skips like this and \vskip below end a paragraph.
{\narrower
```

```
‘‘我们对这一段作了双边缩进的处理。在长段引用时，通常都会这么做。’’
```

```
\bye%如果你愿意的话，可以同时多个段落作如此处理。
```

3:

*italic words***bold words** *put bold and roman inside italic* “倾斜校正” (\/), 使得中间的空白看起来较舒适。这是一些字词---可是标准的 T_EX 字体并不会给出小于 的字体。

如果你需要以下十个字符任意之一:

\$ & # _ % ^ ~ { } \

你需要通过特殊的方式来写他们。参照对开页, 学习正确的记录这些特殊字符的方法。

T_EX 拥有一些语音和字母, 可以用于某些法语词, 例如: *rôle* 以及 *élève*, 或者德语词, 例如 *Schuß*, 或者某些别的什么语言。你可以在第 ``accents'` 页 和 第 ``fornlets'` 页 找到 T_EX 的 欧洲语音和字母表。

你也可以写一些希腊字母, 例如数学中的 ``α`" 以及 ``Ω`", 扑克牌中的 ``♠`" 以及 ``♦`", 一些音乐符号, 例如: ``♯`" 和 ``♭`", 以及一些你可以在 第 ``specsyms'` 页 找到的各种符号。T_EX “数学模式” ``$`

— “起始空白”。具体 的做法可以参看这段的源文件。你可以用1.5取代2 获得 1 1/2空白的效果。(或者可以写成更好看的 $1\frac{1}{2}$ 形式。)

做为举例，我们这里增加了段落的缩进，并且在两段之间增加了一个空行。

```

\xmpheader 4/{行间距}%    ‘xmphead’
\baselineskip = 2\baselineskip % double spacing
\parskip = \baselineskip % Skip a line between paragraphs.
\parindent = 3em % Increase indentation of paragraphs.

% The following macro definition gives us nice inline
% fractions.  You’ll find it in our eplain macros.
\def\frac#1/#2{\leavevmode
  \kern.1em \raise .5ex \hbox{\the\scriptfont0 #1}%
  \kern-.1em $/%$
  \kern-.15em \lower .25ex \hbox{\the\scriptfont0 #2}%
}%

```

~~\frac{1}{2}~~ 起始空曲效果具(本或者可以写成更好看的 $\frac{1}{2}$)

5:

“描述列表”的实例。实际使用时，最好将这些重复结构 定义成宏，并且保证子标题的宽度足够宽。

Queen of Hearts	An ill-tempered woman, prone to saying ``Off with his head!" at the slightest provocation.
Cheshire Cat	A cat with an enormous smile that Alice found in a tree.
Mock Turtle	A lachrymose creature, quite a storyteller, who was a companion to the Gryphon. Reputedly the principal ingredient of Mock Turtle Soup.

这个例子在一个标线盒子中放入一些词句，它们是路易斯·卡罗所写的：

谁不是最想尝一尝，
两便士一碗的好汤？

* * * * *

| 我们给此段落内容添加了修订线效果。修订线用于表示内容改动。

\xmpheader 5/{间隔、标线和盒子}% `'xmphead'`
 %byeqsindent 的 衰 落。 间 隔 随 着 用 户 距 离 的 增 加 而 增 加。 这 些 重 复 结 构

6:

\TeX 5-[p-(Flourosulfonyl)benzoyl]-1, N^6 -ethenoadenosine \TeX “水平盒子溢出”的警告，你可以尝试插入某些“自定连字符”。`\-` 记号告诉 \TeX 一个自定连字符位置；在此位置 \TeX 本来是不会插入连字符的。

你可以不对齐地排版文字，即让右页边不对齐。在过去，文字处理软件还未普及，因为没有其他方便的选择，排版的文档都是不对齐的。有些人更喜欢不对齐地排版文字，因为此时单词间距是一致的。大部分书籍都设置为页边对齐，但并非都是。

27.

这个例子显示如何排版两层的有序列表。如果需要更多层级，唉，你得自己编程。

1. 这是第一个列表项。
2. 这是第二个列表项。它由两个段落组成。为让你看清楚第二个段落从哪里开始，我们缩进了该段落。

第二个段落下边有三个子列表项。

- (a) 这是第一个子列表项。
- (b) 这是第二个子列表项。
- (c) 这是第三个子列表项。
- 这是一个看起来很奇怪的列表项，因为它和其他列表项不同。

这是一个左对齐的行。←

⇒这是一个右对齐的行。

⇒这是一个居中的行。←

```

\xmpheader 6/{杂项}% 'xmphead'
\chardef \ = '\\ % \\
\footline{\hfil{\tenit - \folio -}\hfil}
% \footline
\TeX\
${\it 5}$-[p-(Flouro\ -sul\ -fonyl)ben\ -zoyl]-1,%
$N^6$-ethe\ -no\ -adeno\ -sine
\TeX\%结束符号,警告,个自定义连字符位置;

```

7:

TeX Computer Modern 10p Palatino Roman Palatino Hermann
Zapf Palatino
TeX metafont
Palatino Roman x-height
Metafont Metafont

```
\xmpheader 7/{使用其他来源的字体}%    'xmphead'
\font\tenrm = pplr % Palatino
% Define a macro for invoking Palatino.
\def\pal{\let\rm = \tenrm \baselineskip=12.5pt \rm}
\pal % Use Palatino from now on.
```

```
\TeX Computer Modern
```

```
10p Palatino Roman Palatino
Hermann Zapf
```

```
Palatino
```

```
\TeX\ metafont
```

```
Palatino
Roman
```

```
x-height
```

```
Metafont
```

```
Metafont
```

```
\bye %
```

<i>Pleurotus ostreatus</i>	Grows in shelflike clusters on stumps or logs, pink-gray oyster-shaped caps, stem short or absent.
<i>Lactarius hygrophoroides</i>	Butterscotch-brown cap and stem, copious white latex, often on ground in woods near streams.
<i>Morchella esculenta</i>	Conical cap with black pits and white ridges; no gills. Often found near old apple trees and dying elms in the spring.
<i>Boletus edulus</i>	Reddish-brown to tan cap with yellow pores (white when young), bulbous stem, often near conifers, birch, or aspen.

```

\mpheader 8/{标线表格}% 'xmphead'
\bigskip
\offinterlineskip %
% \tablerule
\def\tablerule{\noalign{\hrule}}
% \tableskip 9pt
\def\tableskip{\omit&height 9pt&&\omit\cr}
% & TeX #
% \bput);
t} \tablerule

```

9:

$$ab\ c\ \alpha\beta\ \gamma$$

$$\cos\alpha=-\cos\beta\cos\gamma+\sin\beta\sin\gamma\cos\alpha\quad(\text{余弦定理})$$

$$\tan\frac{\alpha}{2}=\sqrt{\frac{-\cos\sigma\cdot\cos(\sigma-\alpha)}{\cos(\sigma-\beta)\cdot\cos(\sigma-\gamma)}},\quad\sigma=\tfrac{1}{2}(a+b+c)$$

$$\sin x=\frac{e^{ix}-e^{-ix}}{2i}$$

$$\int_0^\infty\frac{\sin ax\sin bx}{x^2}dx=\frac{\pi a}{2},\quad a<b$$

$$n\ r\ {}_nC_r$$

$$C(n,r) = {}_nC_r = \binom{n}{r} = \frac{n(n-1)\cdots(n-r+1)}{r(r-1)\cdots(1)} = \frac{n!}{r!(n-r)!}$$

$$n\ D$$

$$D=\begin{vmatrix}a_{11}&a_{12}&\cdots&a_{1n}\\a_{21}&a_{22}&\cdots&a_{2n}\\\vdots&\vdots&\ddots&\vdots\\a_{n1}&a_{n2}&\cdots&a_{nn}\end{vmatrix}$$

$$n!$$

$$\sum (\pm)\,a_{1i}a_{2j}\ldots a_{nk}$$

$$i,j,\ldots,k-1\leq n-i,j,\ldots,k+1\leq n$$

$$Q(\xi)=\lambda_1y_1^2\sum_{i=2}^n\sum_{j=2}^ny_ib_{ij}y_j,\qquad B=\|b_{ij}\|=B'$$

10:

$$X\quad |x|$$

$$|x| = \begin{cases} x, & \text{if } x \geq 0; \\ -x, & \text{otherwise.} \end{cases}$$

$$k\geq 0$$

$$x^{k^2} = \overbrace{x\,x\,\cdots\,x}^{2k\text{ times}} \tag{1}$$

$$(2a) \qquad [u][v][w]\,[x]\,[y]\,[z]$$

$$(2b) \qquad u_1'+tu_2''=u_2'+tu_1''$$

$$(2c) \qquad \begin{array}{c} \hat{i} \neq \hat{j} \\ \vec{a} \approx \vec{b} \end{array}$$

$$O(n\log\log n).$$

$$\sum_{i=1}^n x_i = x_1 + x_2 + \cdots + x_n = \text{Sum}(x_1, x_2, \ldots, x_n). \tag{3}$$

$$dx\,dy = r\,dr\,d\theta. \tag{4}$$

$$q\leq 0\quad q$$

$$\{\,q\mid q\leq 0\,\}$$

$$\forall x\exists y\,P(x,y)\Rightarrow\exists x\exists y\,P(x,y)$$

$$P(x,y)\stackrel{\text{def}}{\equiv} x\,\,\,y.$$

```

\xmpheader 10/{更多数学内容}%    'xmphead'
%^{math}
^^{数学}
$X$  $|x|$
$$|x| = \cases{x, &if $x\ge 0$;\cr
-x,&otherwise.\cr}$$

$k \ge 0$

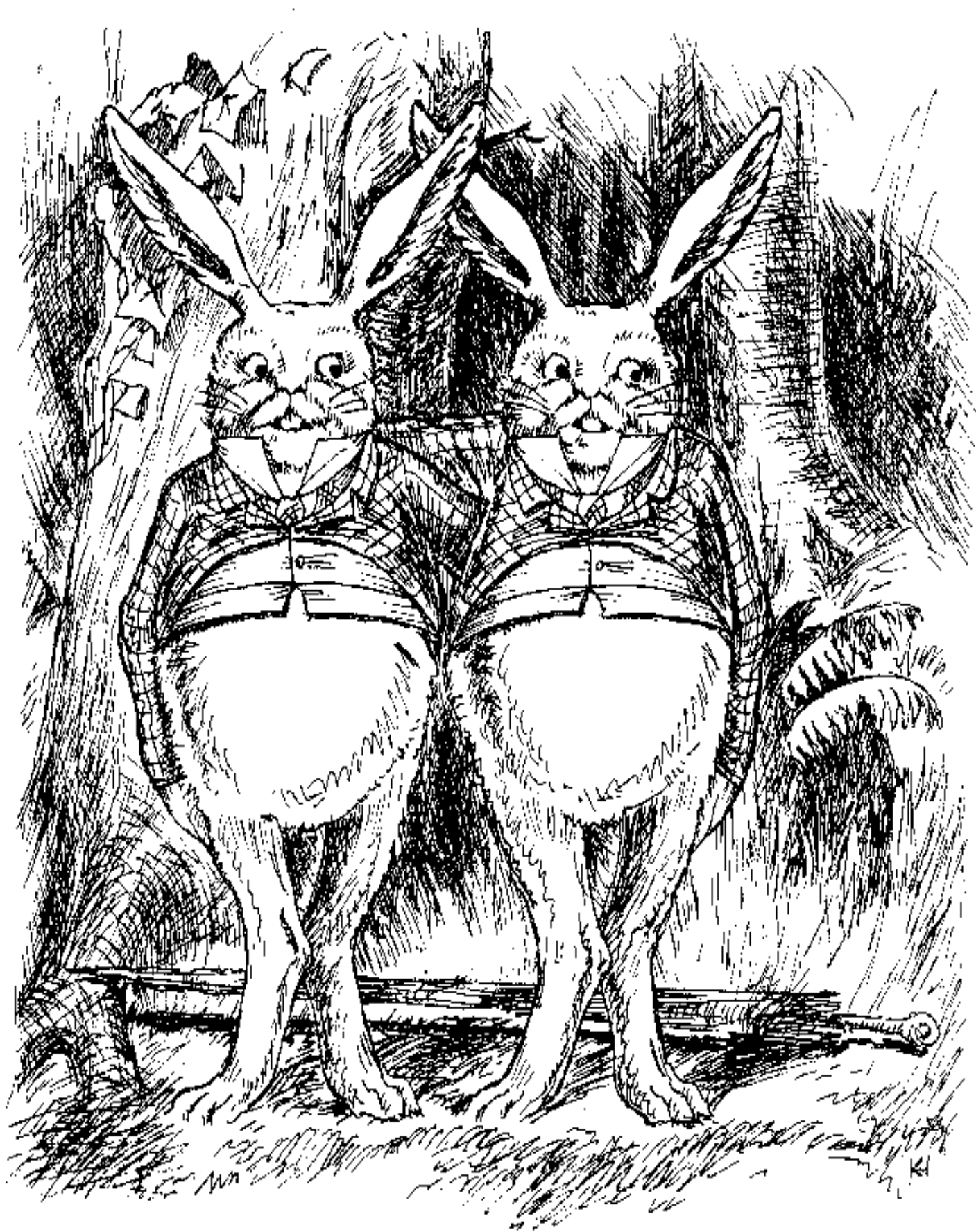
$$x^{k^2}=\overbrace{x\>x\>\cdots\> x}^{2k\ \rm times}$$

\eqno (1)

[u]\![v][w]\,,[x]\>[y]\;;[z]\leqno(2a)

%{\phantom{b}} 一样高的盒子。

```



TeX

The TeXbook

. active character) 是一个有定义的字符。比如一个活动字符可能表示着一个宏定义。你可以把活动字符理解为一种控制序列。当 TeX 碰到了一个活动字符，它就执行这个字符所表示的定义。如果 TeX 碰到的活动字符并没有表示一个定义，则它会报错说这是一个没有定义的控制序列。

一个活动字符的类别码为 13 (`\active` 的值)。当你定义一个活动字符时，你必须先使用 `\catcode` 命令（第 `\catcode` 页）来使这个字符活动，然后再使用类似 `\def`、`\let` 或者 `\chardef` 之类的命令指定这个字符的定义。定义活动字符和定义控制序列所使用的方法是相同的。如因你尝试先对字符指定定义后把它设置成活动的，TeX 会报错说这是一个不明的控制序列。

比如，在 Plain TeX 中，波浪号 (~) 被定义成一个活动字符。它可以在所连接的两个单词当中产生一个空格，并且不让 TeX 把这个空格转为断行。Plain TeX 通过以下方式定义 `~`：

```
\catcode '~ = \active \def~{\penalty10000\~}
```

（这里的 `\penalty` 可以阻止断行，`~`）

. alignment) 是一个把素材有序排印的结构，就如一张有行有列的表格。当你需要构造一个阵列时，你要 (a) 描述行和列的布局，并且 (b) 告诉 TeX 各行或各列中的内容。制表阵列和水平阵列是由一系列的

行组成；竖直阵列则由一系列的列组成。我们首先讲述制表阵列和水平阵列，然后简要地介绍一下竖直阵列。

制表阵列是在 Plain $\text{T}_{\text{E}}\text{X}$ 中定义的。它很简单，因此不如水平阵列那样灵活。制表阵列和水平阵列的主要区别在于你如何定义它们的布局。

要构造一个制表阵列，你首先需要用 `\settabs` 命令（第 `'settabs'` 页）■
确定 $\text{T}_{\text{E}}\text{X}$ 如何将可用空白水平地分为多列。然后你就可以逐行地填充表格的各行。每行由控制序列 `\+`（第 `'@plus'` 页）及其后的一排用 `&` 隔开的“条目”组成（条目是行和列的交叉部分），并以 `\cr` 结束。如果某行的条目数目比该阵列的列数少， $\text{T}_{\text{E}}\text{X}$

```
\settabs \settabs

{\hsize = 1.7 in \settabs 2 \columns
\+cattle&herd\cr
\+fish&school\cr
\+lions&pride\cr}

\settabs 2 \columns 'settabs'  $\text{T}_{\text{E}}\text{X}$  1.7
      cattle      herd
      fish        school
      lions       pride
```

```
{\settabs\+cattle\quad&school\cr
\+cattle&herd\cr
\+fish&school\cr
\+lions&pride\cr}
```

```
      cattle  herd
      fish    school
      lions   pride
```

```
\halign 'halign'  $\text{T}_{\text{E}}\text{X}$   $\text{T}_{\text{E}}\text{X}$  \halign
\halign
```

- # $\text{T}_{\text{E}}\text{X}$ & \cr
- \+ & \cr $\text{T}_{\text{E}}\text{X}$

```
\halign \halign
```

```
\tabskip=2pc
\halign{\hfil#\hfil &\hfil#\hfil &\hfil#\hfil \cr
```

```

&&\it Table\cr
\noalign{\kern -2pt}
\it Creature&\it Victual&\it Position\cr
\noalign{\kern 2pt}
Alice&crumpet&left\cr
Dormouse&muffin&middle\cr
Hatter&tea&right\cr}

```

	<i>Table</i>
<i>Creature</i>	<i>Victual</i>
Alice	crumpet
Dormouse	muffin
Hatter	tea
	<i>Position</i>
	left
	middle
	right

\tabskip ‘`\tabskip`’ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 2pc \noalign ‘`\noalign`’ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ \noalign Table” 和 “Position” 紧挨在一起。（在首行之前或者末行之后你同样可以使用 \noalign 命令。）

用 \valign 命令（第 ‘`\valign`’ 页）可以构造一个竖直阵列。竖直阵列按照列而不按照行组织。它遵循和水平阵列相同的法则，只是此时行和列的地位交换了。例如，下面的竖直阵列：

```

{\hsize=0.6in \parindent=0pt
\valign{#\strut&#\strut&#\strut\cr
one&two&three\cr
four&five&six\cr
seven&eight&nine\cr
ten&eleven\cr}}

```

得到的结果为：

one	four	seven	ten
two	five	eight	eleven
three	six	nine	

模板中的 \strut 命令（第 ‘`\strut`’ 页）是必不可少的，因为它让一行的各条目合适地排成一行，即让它们有共同的基线，并让基线间的距离保持一致。

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$. The $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ book $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

- $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ \input ‘`\input`’ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$
- $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ The $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ book 46-47 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

- `\expandafter` `'expandafter'`
- `\input` `'input'` `\catcode` `'catcode'`
- `\input`

. argument) 包含传递给命令的文本, 它补全了该命令正常运行所需的信息。这里的命令可以是原始命令或者宏。

原始命令的参量有各自规定的形式。例如, 下面的一系列记号:

```
\hskip 3pc plus 1em
```

由 `\hskip` 命令及其参量 `3pc plus 1em` 组成。但若你这样写:

```
\count11 3pc plus 1em
```

你将得到完全不同的结果。TeX 将认为 `\count11` 为命令, 其参量为 `'3'`, 而后面的 `pc plus 1em` 是普通的文本记号 (因为计数寄存器要求一个整数值作为参量)——这多半不是你想要的。顺便说一下, 该命令将给第11号计数寄存器赋值3 (见 `\count` 这里的讨论, 第 `'count'` 页)。

另一方面, 所有宏都遵循相同的参量约定。传递给宏的每个参量都对应宏定义中的一个参数。宏的参数可以是“定界的”或是“非定界的”。在宏的定义中确定了宏参数的个数和类型, 从而确定了宏参量的个数和类型。

定界参量和非定界参量的区别在于, TeX 用不同方式确定该参量的结束位置。

- 定界参量由该参量开始处和作为该参量定界子的特定记号序列之间的所有记号组成, 但不包含该定界子序列。其中的定界子是在宏的定义中规定的。因此要给出定界参量, 你需要写上该参量并在其后加上定界子。定界参量也可以是空的, 即不包含任何文本。定界参量里面的任何花括号都必须配好对, 即每个左花括号必须有对应的右花括号, 反之亦然。
- 非定界参量由单个记号或者包含在花括号的一系列记号组成, 比如 `{Here is {the} text.}`。外面的一对花括号, 虽然看起来像, 实际上并不组成一个编组——TeX 只是用它们确定参量的范围。而参量内部的花括号, 比如 `{the}` 两边的, 必须正确配对。若你误将太多右花括号放在里面, TeX 将报错说有多余的右花括号。若左花

括号太多， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 同样会报错；但可能出现在该参量所期望的结束位置之后很远的地方（见第 `'mismatched'` 页）。

请参阅 “” depth height (第 `'` 页) 这里的更多关于参数和参量的内容。在 *The $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ book* 第 203--204 页你能找到定界参量和非定界参量的精确规则。

ASCII. ASCII American Standard Code for Information Interchange) 的缩写。总共有 256 个 ASCII 字符，每个都有自己的编码，但只有前面 128 个是标准编码。在 ASCII “编码表”，比如 *The $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ book* 第 367 页里面，你可以找到这些编码的含义。编码在 32--126 之间的字符为“可打印字符” depth height，比如字母，数字和标点符号。其余字符为“控制字符” depth height，通常用于（在计算机工业中，而不是 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 中）控制输入/ ASCII 84 ‘T’，而 ASCII 码 12 对应于“换页”功能（大多数打印机都将它解释为“开始新页”）。虽然 ASCII 标准指定了控制字符的含义，但很多设备，比如调制解调器和打印机，都将控制字符用于标准规定之外的目的。

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 中字符的含义通常和标准 ASCII 中的含义一致。而且在包含 ASCII 可打印字符的字体中，这些字符所在的位置也和 ASCII 中的一样。但有些字体，特别是数学字体，将 ASCII 可打印字符替换为与 ASCII 无关的字符。比如，计算机现代数学字体 `cmsy10` 在 ASCII 数字 `'8'` 的位置上是数学符号 `'\forall'`。

. assignment) 是一种为 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 的寄存器，内部参数，内部表格项，或者控制序列分配值的结构。赋值的一些例子如下：

```
\tolerance = 2000
\advance\count12 by 17
\lineskip = 4pt plus 2pt
\everycr = {\hskip 3pt \relax}
\catcode\('@ = 11
\let\graf = \par
\font\myfont = cmbx12
```

第一个赋值要求 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 分配数值 2000 给数值参数 `\tolerance`，即让 `\tolerance` 的值等于 2000。其他赋值类似。赋值中的 `'='` 和空格是可省略的，因此你也可以把第一个赋值改写为下面更紧凑的形式：

```
\tolerance2000
```

在 *The $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ book* 第 276--277 页中对赋值语法有详细介绍。

. badness) 值测量该行的字间隔偏离它们的自然值多远, 其中自然值就是文本行所用字体中指定的值。偏离越大, 劣度就越大。类似地, 页面的劣度测量页面盒子的间隔偏离它们的理想值多远。(一般地, 这些盒子大多数是段落的文本行。)

更准确地说, 劣度测量这些间隔对应的粘连要伸缩多少才能正好填满文本行或页面。T_EX 让劣度近似等于, 为组成所需尺寸的行或页, 粘连必须伸缩的比例的立方的 100 倍。例如, 粘连伸长了给定伸长量的两倍得到的比例为 2 而劣度为 800; 粘连伸长了给定伸长量的一半得到的比例为 .5 而劣度为 13。T_EX 将超过 10000 的劣度视为等于 10000。

T_EX 分段为行时使用行劣度 (见 “” depth height, 第 ‘l’ 页)。它在下面两个步骤中使用行劣度:

- 1) 在选择断行点时, T_EX 要求各行劣度小于或等于 `\tolerance` 值 (第 ‘tolerance’ 页)。如果不得不放置一个劣度超过此值的行, T_EX 将把它设定为未滿或过滿水平盒子。T_EX 仅在万不得已时才会放置未滿或过滿水平盒子, 即仅在没有其他方法分段为行时。
- 2) 假设各行都相当糟糕, T_EX 用行劣度评价不同的分段为行方式。在评价过程中它对每个可能的行结合一个 ‘缺陷’。劣度会增加缺陷值。然后 T_EX 选择让段落总缺陷最小的方式来分段为行。在多数情况下, T_EX 使用让最糟糕行的劣度最小的方式安排段落。请参阅 *The T_EXbook* 第 97--98 页中 T_EX 如何分段为行的细节。

T_EX 将一系列文本行和其他竖直模式素材组装为页面的过程与断行的过程类似。但是组装页面没那么复杂, 这是由于 T_EX 寻找分页点时每次只考虑一个页面, 因此只需要决定在哪里结束当前页面。与此相反, 当选择断行点时 T_EX 要同时考虑多行。(大多数文字处理器每次只选择一个断行点, 因而断行做得不如 T_EX。) 请参阅 *The T_EXbook* 第 111--113 页中 T_EX 如何选择分页点的细节。

. baseline) 是一条想象中的横穿盒子的线。在将水平列表的盒子组装为更大的盒子时, T_EX 将列表中各盒子对齐, 以让它们的基线重合。打个比方, 想象你在标线本上写字, 你所写的每个字母都有一条隐含基线。为让各字母水平对齐, 你会让它们的基线和本子上印刷的浅色提示线重合。

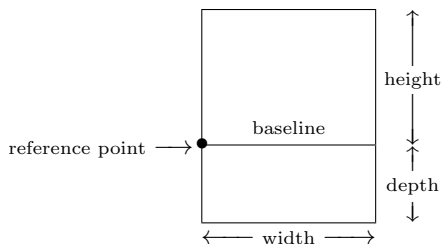
盒子可以而且也经常会延伸到基线下边。比如, 字母 ‘g’ 就延伸到其盒子的基线下边, 因为它有一个降部 (‘g’)

. box) 是要排版的矩阵素材。单个字符自身就是一个盒子, 整个页面同样也是一个盒子。T_EX 用盒子套盒子套盒子以形成页面。最外层盒

子就是页面自身，最内层盒子多半是单个字符，而单个文本行的盒子在中间某个地方。

\TeX 在构造段落和页面时隐式执行大多数盒子构建活动。利用一些 \TeX 命令，比如 \hbox （第 [‘\hbox’](#) 页）、 \vbox （第 [‘\vbox’](#) 页）和 \vtop （第 [‘\vtop’](#) 页），你可以显式构造盒子。 \hbox 命令通过从左到右水平添加较小的盒子构造一个盒子；它对一个水平列表进行操作以生成一个水平盒子（horizontal box）。 \vbox 和 \vtop 命令通过从上到下竖直添加较小的盒子构造一个盒子；它们对一个竖直列表进行操作以生成一个竖直盒子（vertical box）。这些水平列表和竖直列表不仅可以包含较小的盒子，还可以包含其他几种实体，比如粘连和紧排。

盒子有高度、深度和宽度，如下所示



基线（baseline）如同标线本上的浅色提示线。像 $\text{\texttt{g}}$ 这样的字母盒子延伸到基线下边；而像 $\text{\texttt{h}}$ 这样的字母盒子却不会。盒子的高度是盒子在基线上边的延伸距离，而它的深度是在基线下边的延伸距离。盒子的基准点（reference point）是它的基线与左边缘的交点。

在从水平列表构建水平盒子 H 时， \TeX H H H ¹ H H \TeX
 H *The \TeX book* 77

V \TeX H V V ² V \TeX V V

\vbox \vtop V V

▪ $\text{\vbox}\text{\TeX}$ V \TeX V ³

▪ $\text{\vtop}\text{\TeX}$ V \TeX V

\vbox \vtop \vtop \vbox *The \TeX book* 78 80–81 \TeX

\newbox ‘@newbox’ ‘

. category code) 决定它在 \TeX 中扮演的角色。例如， \TeX \TeX
 \TeX ‘r’ 11 \TeX

¹ \raise \lower \TeX

² \moveleft \moveright \TeX

³ \boxmaxdepth ‘boxmaxdepth’

`\TeX \TeX \TeX \text{expouxi} \char \char \char \TeX \backslash` (通常的转义字符) 的 ASCII 码是 92。如果你键入 `\char92 grok`, 却等同于 `\grok`。实际上它让 `\TeX` 排版出 `\cgrok`, 其中 `c` 是当前字体的编码表中位置为 92 的字符。

你可以用 `\catcode` 命令 (第 `\catcode` 页) 重新设定任何字符的类别码。改变了类别码就改变了各种字符所扮演的角色。例如, 如果你键入 `\catcode\@ = 11`, `@` 符号的类别码就被设定为“字母符”。这样你就可以在控制序列名称中使用 `\@`。

这里给出 plain `\TeX` 定义的类别码列表, 以及每个类别中的字符 (第 `\twocarets` 页有 `^^` 记法的解释):

0	转义符	<code>\</code>
1	编组开始符	<code>{</code>
2	编组结束符	<code>}</code>
3	数学环境符	<code>\$</code>
4	阵列制表符	<code>&</code>
5	行结束符	<code>^^M</code> \equiv ASCII \langle return \rangle
6	宏参数符	<code>#</code>
7	上标符	<code>^</code> 和 <code>^^K</code>
8	下标符	<code>_</code> 和 <code>^^A</code>
9	可忽略符	<code>^^@</code> \equiv ASCII \langle null \rangle
10	空格符	<code>␣</code> 和 <code>^^I</code> \equiv ASCII \langle horizontal tab \rangle
11	字母符	<code>A ... Z</code> 和 <code>a ... z</code>
12	其他字符	(不属于其他类别的字符)
13	活动字符	<code>~</code> 和 <code>^^L</code> \equiv ASCII \langle form feed \rangle
14	注释符	<code>%</code>
15	无效字符	<code>^^?</code> \equiv ASCII \langle delete \rangle

除了第 11--13 类别, 一个类别中的所有字符产生的效果都相同。例如, 假设你键入:

```
\catcode\[ = 1 \catcode\] = 2
```

则左方括号和右方括号分别就变成编组开始符和编组结束符, 即它们分别和左花括号和右花括号等价。在这些定义之下, `{[a b]}` `{a b}` `{a b}`

```
11 12
13 \TeX
\TeX 15
'^^K' '^A' 7 8 ASCII \TeX
```

```

TEX TEX TEX

\def\foo{\catcode'\$ = 11 } % $
\foo$ % '$'。
\foo$ % 未定义的控制序列 `foo$'。

```

T_EX '\$'。在第二行，当 T_EX 首次查看 '\$' 时，它是在寻找控制序列名称的结束位置。由于 '\$' 还不是字母符，它表示 '\foo' 已结束。接下来，T_EX '\foo' '\$' 11 T_EX '\$'。由于 '\$' 已经是字母符，T_EX '\$' T_EX '\$' \foo\$

```

TEX \fum \fum ℓ TEX ℓ \fum

```

· T_EX character): 作为读取的输入字符，以及作为排版的输出字符。T_EX 将大多数输入字符转变成输出字符以显示它们。例如，它通常将输入字母 'h' 转换为用当前字体排版的字母 'h'。但是，对于类似 '\$' 这些有特别意义的字符，却不是这么回事。

T_EX 通过读取输入文件（或终端）和展开宏得到输入字符。它们是 T_EX 获取输入字符的途径。每个输入字符有一个编码，它表示该字符在 ASCII 码表的位置。例如，字母 'T' 的 ASCII 编码为 84。

在读取字符时，T_EX 给它附加一个类别码。类别码影响 T_EX 读取字符时对该字符的解释。在读取宏定义时，T_EX T_EX 'texpoux' T_EX The T_EXbook 46–48

ASCII 0–31 127–255 T_EX T_EX '^c'，其中 *c* 为任何字符，你将得到一个比 *c* 的 ASCII 码大 64 或小 64 的字符。这种表示法得到的编码不能超过 127，因此它是无歧义的。这种表示法的三个最常见例子是 '^M' (ASCII <return> 字符)，'^J' (ASCII <line feed> 字符) 和 '^I' (ASCII <horizontal tab> 字符)。

T_EX 还有另一种指明 ASCII 编码的表示法，它能够表示从 0 到 255 的所有字符。如果你键入 '^xy'，其中 *x* 和 *y* 是任何十六进制数字 '0123456789abcdef'，你就得到该编码对应的字符（这里只能使用小写字母。）T_EX 优先选择“十六进制数字”这种解释，因此在类似 '^a' 的字符后面不要跟着一个小写十六进制数字——这样将会得到错误的解释。在需要使用这种表示法时，你会发现有个 ASCII 编码表更方便些。

输出字符就是待排版的字符。生成输出字符的命令有这样的意义：“生成一个包含当前字体第 *n* 个字符的盒子”，其中 *n* 由该命令给出。T_EX 将这些盒子和其他排版元素组合起来并在页面上排好，以生成你排版的文档。

类别码为 11（字母）或 12（其他）的输入字符，是一个生成对应输出字符的命令。另外，你也可以用 '\char *n*'（第 '\char' 页）让

\TeX 生成字符 n ，其中 n 是介于 0 和 255 之间的数。命令 `\h'`、`\char'h` 和 `\char104` 的结果相同（104 是 `\h'` 的 ASCII 编码。）

· `class`) 指明该字符在数学公式中的角色。字符的类记录在它的数学码中。例如，等号 `\=` 属于第 3 类（关系符号）。 \TeX 利用字符类的知识确定数学公式的各部分的间隔的大小。例如，下面这个公式首先用正常方式显示，然后用随机修改各字符类的方式显示：

$$a + (b - a) = a \qquad a + (b - a) = a$$

参见本书第 `\mathord'` 页中各种类的列表，以及 *The \TeX book* 第 154 页中各种类的含义。

· `command`) 指导 \TeX 执行某种动作。到达 \TeX \TeX `'texpouxi'`

\TeX \TeX
`\vskip 1in` \TeX 1 `'1in'` `\vskip`
 \TeX

- `'W'` \TeX `'W'`
- `\bf`
- `\'` `'è'`
- `\P¶\aeæ)`
- 参数，比如 `\parskip` 说明 \TeX 在段落间添加的粘连的大小
- 数学符号，比如 `\alpha` (α) 和 `\in` (\in)
- 数学运算符，比如 `\over` 生成一个分式

· `conditional test`) 是一个命令，它测试某个条件是否成立，并依此让 \TeX 展开或忽略某些文本。条件测试的一般形式是：

`\if α (true text)\else(false text)\fi`

或者：

`\if α (true text)\fi`

其中 α 指定某个特定的测试。例如，`\ifvmode` 测试 \TeX 是否位于竖直模式中。如果条件成立， \TeX `<true text>` \TeX `<false text>` \TeX \TeX `'texpouxi'` `'conds'`

· control sequence) 是 $\text{T}_{\text{E}}\text{X}$ 命令的名称。控制序列总是以转义符开头, 即以反斜杠 (\) depth height plus 0.1em minus 0.1 em 开头。控制序列必为如下两种形式中的一种:

- 控制词是由转义符紧跟一个或多个字母组成的控制序列。控制词在 $\text{T}_{\text{E}}\text{X}$ 碰到非字母符时结束。例如, 当 $\text{T}_{\text{E}}\text{X}$ 读取 `\hfill, the` 时, 它看到六个记号: 控制序列 `\hfill`、逗号、空格、`t`、`h`、`e`。在 $\text{T}_{\text{E}}\text{X}$ 扫描控制序列时, `\hfill` 之后的空格结束该控制序列并被吸收掉。(另一方面, 对于 `\hfill, the`, 逗号不仅结束控制序列, 而且自身也算作一个字符。)
- 控制符是由转义符紧跟单个非字母符组成的控制序列——非字母符也可以是空格符或者行尾符。控制符是自定义的, 即 $\text{T}_{\text{E}}\text{X}$ 无需读取后面的字符就知道它在哪里结束。控制符之后的字符永远不会被吸收掉。

你可以参考第 [spaces](#) 页, 其中有对控制序列后的空格的介绍。

$\text{T}_{\text{E}}\text{X}$ 提供了许多预定义的控制序列。原始的控制序列内建于 $\text{T}_{\text{E}}\text{X}$ 程序中, 因此可以在各种 $\text{T}_{\text{E}}\text{X}$ 形式中使用。其他预定义的控制序列由 plain $\text{T}_{\text{E}}\text{X}$ (即本书描述的 $\text{T}_{\text{E}}\text{X}$ 形式) 提供。

你可以用自己的控制序列扩充预定义的控制序列, 只需用 `\def` 和 `\let` 等命令定义它们。本书第 [explain](#) 章包含了一批实用的控制序列定义。此外, 你的计算设备也可能提供一批本地开发的 $\text{T}_{\text{E}}\text{X}$ 宏。

· control symbol) 是由转义符紧跟单个非字母符组成的控制序列——非字母符也可以是空格符或者行尾符。

· control word) 是由转义符紧跟一个或多个字母组成的控制序列。⁴ $\text{T}_{\text{E}}\text{X}$ 忽略控制词之后的空格符或行尾符, 仅仅用它们表示控制词的结束。

· [math](#)

· delimiter) 是一个作为数学公式的可见边界的字符。 $\text{T}_{\text{E}}\text{X}$ 可以根据子公式的竖直尺寸 (高度加深度) depth height plus 0.1em minus 0.1 em 调整定界符大小, 这就是定界符的本质特性。然而, 要让 $\text{T}_{\text{E}}\text{X}$ 对定界符作调整, 定界符必须出现在“定界环境”中, 即作为 `\left`、`\right`、`\overwithdelims`、`\atopwithdelims` 或 `\abovewithdelims` (见 [overwithdel](#)) 这些命令之一的参量。定界环境也包括在定界环境中使用参量的宏的任何参量。

⁴ 11

- 2) 你可以用 `\delimiter` 命令 (第 `'\delimiter'` 页) 直接生成一个定界符, 这与你用 `\char` 命令生成一个普通字符或者用 `\mathchar` 命令生成一个数学字符类似。 `\delimiter` 命令使用的定界码和 `\delcode` 表格项用到的一样, 但前面多了一个数字用于标示所属的类。除了在宏定义中, 其他地方很少用到 `\delimiter` 命令。

定界码 (delimiter code) 告诉 $\text{T}_{\text{E}}\text{X}$ 如何搜索合适的输出字符来表示一个定界符。这些搜索规则相当复杂 (见 *The $\text{T}_{\text{E}}\text{X}$ book* 第 156 和 442 页)。要完全理解这些规则需要熟悉字体度量文件的结构, 而这个主题不仅超出本书范围, 也超出 *The $\text{T}_{\text{E}}\text{X}$ book* 的范围。

本质上搜索过程是这样运行的。定界码给出了字体族和字体位置, 从而指定了一个“小号”输出字符和一个“大号”输出字符 (见第 `'\delcode'` 页)。利用这些信息, $\text{T}_{\text{E}}\text{X}$ 能够找到 (或者构造) 定界符的越来越大的版本。 $\text{T}_{\text{E}}\text{X}$ 首先尝试“小号”字体的“小号”字符的不同尺寸 (从小到大), 然后是“大号”字体的“大号”字符的不同尺寸 (同样是从从小到大), 寻找一个高度加上深度足够大的字符。如果所找到的字符都不够大, 它就用最大那个。也有可能小号字符或大号字符, 或两者都没指定 (即定界码的对应部分为零)。如果只指定了一个字符, $\text{T}_{\text{E}}\text{X}$ 使用那一个。如果两个字符都没指定, 它就将定界符替换为宽度为 `\nulldelimiterspace` 间隔。

· $\text{T}_{\text{E}}\text{X}$ demerits) 度量它分段为行时文本行的不良度 (见 `"` depth height, 第 `'` 页)。文本行的缺陷同时受该行的劣度和它带有的惩罚影响。 $\text{T}_{\text{E}}\text{X}$ 的目标是合理地分配各行文本, 以让段落的总缺陷最小, 其中总缺陷由各行的缺陷加起来。参见 *The $\text{T}_{\text{E}}\text{X}$ book* 第 97--98 页以了解 $\text{T}_{\text{E}}\text{X}$ 分段为行的细节。 $\text{T}_{\text{E}}\text{X}$ 选择分页点时不使用缺陷; 它使用的是被称为分页“代价”的类似概念。

· depth) 是盒子在基线之下的距离。

· dimension) 指定了一个距离, 即对空间的长度测量。你用尺寸指定事物的大小, 比如文本行的长度。英语国家的打印机习惯用点 (point) 和派卡 (pica) 来测量距离, 而欧洲大陆的打印机习惯用迪多点 (didot points) depth height plus 0.1em minus 0.1 em 和西塞罗 (cicero) depth height。你可以使用这些单位, 或者其他你更熟悉的单位, 比如英寸。 $\text{T}_{\text{E}}\text{X}$ 能够识别的与字体无关的度量单位有如下这些:

pt	point	点	72.27 points = 1 inch
pc	pica	派卡	1 pica = 12 points
bp	big point	大点	72 big points = 1 inch

in	inch	英寸	
cm	centimeter	厘米	2.54 centimeters = 1 inch
mm	millimeter	毫米	10 millimeters = 1 centimeter
dd	didot point	迪多点	1157 didot points = 1238 points
cc	cicero	西塞罗	1 cicero = 12 didot points
sp	scaled point	缩点	65536 scaled points = 1 point

还有两个与字体相关的度量单位：`ex` 是一个纵向尺寸，通常与字体中字母 `x` 的高度相关；`em` 是一个横向尺寸，通常等于字体的大小，与字体中字母 `M` 的宽度相关。最后，TeX ‘fil’, ‘fill’ ‘filll’, 它们的强度的阶依次增大。

尺寸可以用一个因子，即乘数，后面加上度量单位来表示。因子可以是一个整数，或是一个带有小数点或小数逗号的小数。因子之前可以带有加号或减号，因此尺寸可正可负。即使数值为零，度量单位也必须写上。数值和度量单位之间可以有也可以没有空格。在*The TeXbook* 第 270 页中有尺寸的精确定义。下面这些是尺寸的例子：

```
5.9in    0pt    -2,5 pc    2fil
```

其中最后一个尺寸表示一个一阶无限距离。

无限距离远远大于任何有限距离或者任何低阶的无限距离。如果你给 `.001fil` 加上 `10in`，你得到 `.001fil`；如果你给 `-1fill` 加上 `2fil`，你得到 `-1fill`，依此类推。只有在指定粘连的伸长量和收缩量时，才可以使用无限距离。

TeX 对文档中的所有尺寸都乘以一个放大率因子 $f/1000$ ，其中 f 是 `\mag` 参数的值。由于 `\mag` 的默认值为 1000，在一般情形下文档是照常排版的。你可以指定一个与放大率无关的最终文档尺寸，只要在度量单位前加上 `true`。例如，`\kern 8 true pt` 8

. `display math`) 表示在单独一行显示的公式；TeX 在陈列公式上边和下边添加额外间隔，以隔开周围文本。陈列公式两边用 `$$` 括起来。TeX 在陈列数学模式中读取陈列公式。

. `escape character`) 引入一个控制序列。Plain TeX 的转义符是反斜杠 (`\`)。利用 `\catcode` 命令 (第 `'catcode'` 页) 重新定义 c_1 和 c_2 的类别码，你可以将转义符从 c_1 改为 c_2 。类似地你可以定义额外的转义符。如果你想排版包含字面转义符的材料，你必须或者 (a) 定义一个表示打印转义符的控制序列，或者 (b) 使用第 `'verbatim'` 页介绍的方法，改变转义符的类别码以临时取消转义。下述定义：

```
\def\{\{$\backslash$}
```

是一种生成表示 `\'` 的控制序列的方法（用数学字体排版反斜杠）。

对于合成控制序列，比如那些由 `\string` 和 `\message` 生成的控制序列，你可以用 `\escapechar` 参数（第 `\escapechar` 页）指定其中的转义符用哪个字符表示。

`\family`）是用于数学模式的，由三个相关字体组成的集合。在数学模式之外，族没有任何作用。一个族的三个字体分别用于正常符号（文本尺寸）`depth height`，上标和下标（标号尺寸）`depth height`，以及次上标和次下标等（小标号尺寸）`depth height`。例如，数字 `'2'` 用这三个字体分别显示为 `'2'`、`'_2'` 和 `'^2'`（在 plain `TEX` 中）。通常你会将一个族的三个字体设定为相同字型且不同点数，但没人阻止你将这三个字体也设定为不同字型，或者将其中两个字体设定为完全一样。

`TEX` 提供多达十六个字体族，它们用 0–15 编号。比如在 plain `TEX` 中，第 0 族由用于文本尺寸的 10 点罗马字体，用于标号尺寸的 7 点罗马字体，以及用于小标号尺寸的 5 点罗马字体组成。Plain `TEX` 还将第 1 族定义为由数学意大利字体组成，并将第 2 族和第 3 族分别保留给特殊符号和数学扩展符号。^{7 8} 如果你需要自己定义字体族，你应该使用 `\newfam` 命令（第 `\@newfam` 页）获取未使用的族编号，再用 `\textfont`、`\scriptfont` 和 `\scriptscriptfont` 命令（第 `\scriptscriptfont` 页）分别设定该族的各个字体。

`\file`）是 `TEX` 解释或者创建的信息流。文件由监督 `TEX` 运行的操作系统管理。`TEX` 在四种不同的背景中用到文件：

- 1) “源码文件”是 `TEX` “眼睛”读取（见“`TEX`” `depth height`，第 `\texpouxi` 页），并根据它的一般法则解释的文件。你的主要输入文件——即调用 `TEX` 时在 `'**'` 之后或者在命令行中指定的文件——是一个源码文件，你用 `\input` 命令（第 `\input` 页）请求的任何文件同样也是。
- 2) “结果文件”是包含 `TEX` 运行结果的文件。`TEX` 运行后产生两个结果文件：`.dvi` 文件和日志文件。`.dvi` 文件包含打印文档所需的信息，而日志文件包含运行记录，包括 `TEX` 生成的任何错误信息。若主要输入文件的名称是 `screed.tex`，则 `.dvi` 文件和日志文件的名称分别是 `screed.dvi` 和 `screed.log`。⁹
- 3) 要用 `\read` 命令（第 `\read` 页）读取一个文件，你需要将这个文件和一个输入流关联起来。你最多可以有 16 个同时活动的输入

⁷ 2 3

⁸ 4–7 `\itfam\slfam\bffam\ttfam`

⁹ `TEX`

流，它们以 0-15 编号。 `\read` 命令读取一行并把它作为指定控制序列的值，因此用 `\read` 读取和用 `\input` 读取是大不相同的（后者读入整个文件）。`TEX` 将编号不在 0 和 15 之间的输入流视为终端，因此，比如说 `\read16`，将读取你在终端中键入的下一行文本。

- 4) 要用 `\write` 命令（第 `'write'` 页）写入一个文件，你需要将这个文件和一个输出流关联起来。你最多可以有 16 个同时活动的输出流，它们以 0-15 编号。输入流和输出流是相互独立的。任何送到编号为负数的输出流的东西，将被写到日志文件中；而任何送到编号大于 15 的输出流的东西，将同时被写到日志文件和终端中。因此，比如说 `\write16`，将在终端中写上一行文本并将它送到日志文件中。

在使用流文件之前，你必须先打开它。输入流文件用 `\openin` 命令（第 `'openin'` 页）打开，而输出流文件用 `\openout` 命令（第 `'openout'` 页）打

```
TEX \closein 'closein' \closeout 'closeout'
      \newread \newwrite '@newwrite'
TEX \shipout The TEXbook 227 \write \immediate 'immediate' TEX
```

```
\immediate\write16{Do not pass GO! Do not collect $200!}
```

· file name) 是文件在监督 `TEX` 运行的操作系统中的名称。文件名的语法遵循 `TEX` 语法的一般规则，而且实际上在不同的 `TEX` 实现中它们各不相同。特别地，大多数 `TEX` 实现都认为文件名在空格符或行尾符处结束。因此 `TEX` 有可能曲解 `{\input chapter2}`，将右括号视为名称的一部分。一般来说，你应该在文件名后面添加一个空格符或行尾符，就像 `{\input chapter2 }` 这样。

· `TEX font`) 是由不超过 256 个输出字符组成的集合，这些字符通常有相同的设计、样式（罗马体、意大利体、粗体、窄体等）和点数。¹⁰ `TEX` 自带的计算机现代字体一般只有 128 个字符。在本书最后面的书尾页中描述了排版本书所用的字体。

举个例子，这里是 10 点 Palatino 罗马字体的字母表：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

¹⁰ Plain `TEX` logo

而这里是 12 点计算机现代粗体扩展字体的字母表：

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

字体中的字符都是编好号的。对于在 ASCII 字符表中存在的字符，它们的编号一般与其 ASCII 编码一致。每个字体的编码表指明该字体的第 n 个字符是怎样的。有些字体，比如那些用于数学符号的字体，根本不包含任何字母字符。你可以用 `\char n`（第 `\char` 页）生成包含第 n 个字符的盒子，该字符用当前字体排版。

要在文档中使用某个字体，你必须先用控制序列给它命名再载入它。此后，在需要用到它的地方，你就可以键入该控制序列来选择它。Plain TeX 提供了若干已经命名和载入的字体。

你可以用像 `\font\twelvebf=cmbx12` 这样的命令在同一个操作中命名并载入字体。这里的 `\twelvebf` 是用于命名该字体的控制序列，而 `cmbx12` 表示计算机文件系统中的字体度量文件 `cmbx12.tfm`。现在你可以通过键入 `\twelvebf` 开始使用这个字体。在此之后，这个字体将一直生效，直到 (a) 你选择了另一个字体，或者 (b) 你结束了编组，如果在开始字体时就位于这个编组中。例如下面的输入：

```
{\twelvebf white rabbits like carrots}
```

将使得 `cmbx12` 字体只对文本 `white rabbits like carrots` 生效。

在 TeX 中你可以使用与计算机现代不同的字体（见第 `\palatino` 页的例子以及页眉）。这些字体的文件需要安装在计算机文件系统的某个 TeX 能找到的位置中。对每个字体，TeX `cmbx12.tfm cmbx12.pk`

TeX

`cmbx12 cmbx12.pk cmbx12.gf`

. footer) 是放在每个页面底部，位于正文之下的素材。在 plain TeX 中默认页脚是居中的页码。通常页脚只包含一行文本，你可以通过给 `\footline`（第 `\footline` 页）指定一个记号列来设定页脚。请参阅第 `\bighead` 页中介绍的生成多行页脚的方法。

. format file) 是一个包含 TeX 的内存映像的文件，该文件以一种能够快速重新载入的形式存储。格式文件可以用 `\dump` 命令（第 `\dump` 页）建立。映像文件包含转储发生时存在的（字体和宏等）定义的完整记录。然后利用 `virtex`, TeX TeX TeX

`initex initex TeX TeX virtex initex`

\TeX \tex \TeX \plain \TeX \Plain \TeX \TeX \tex \virtex \plain \TeX
 \TeX

. \global) 定义在文档结束前始终有效，即使它出现在一个编组里面 (除非它被其他定义覆盖)。即全局定义不受编组范围的影响。要让任何定义成为全局的，你可以在它前面加上 \global 命令 (第 \global 页)，除非 \globaldefs (第 \globaldefs 页) 为负值。

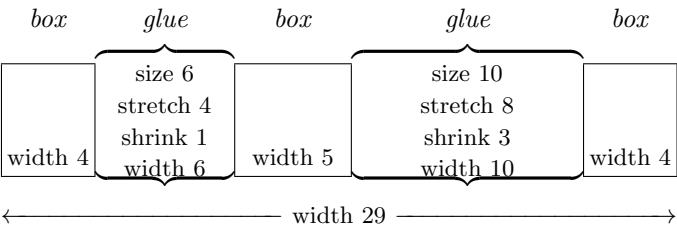
还有种特殊方法将一个宏定义为全局的。通常你用 \def 命令或者 \edef 命令 (第 \edef 页) 来定义一个宏。如果你将 \def 和 \edef 分别换为 \gdef 和 \xdef ，这个宏定义就会是全局的。也就是说， \gdef 等价于 \global\def ，而 \xdef 等价于 \global\edef 。

. \glue) 是可以伸长或者收缩的空白间距。粘连给 \TeX 提供了生成漂亮文档所需的灵活性。粘连有两种类型：水平粘连和竖直粘连。水平粘连出现在水平列表中，而竖直粘连出现在竖直列表中。你可以隐式地生成一个粘连项，比如用单词间空格，或者显式地生成一个粘连项，比如用 \hskip 命令。在排版文档时 \TeX 本身也生成很多粘连项。这里我们只描述水平粘连——竖直粘连与之类似。

在 \TeX 将一系列盒子和粘连组装成一个较大单元时，它调整各个粘连的尺寸以适合较大单元的空间要求。例如，为了让页面的右边距保持一致， \TeX

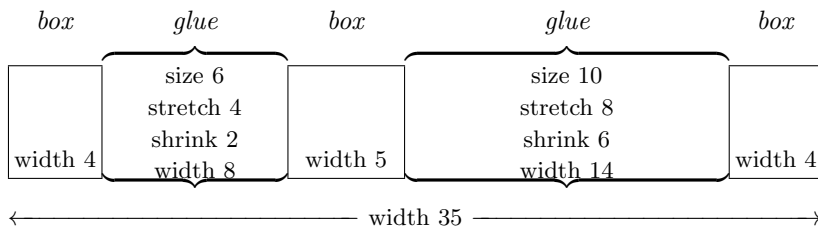
\stretch) 和它的收缩量 (\shrink)。你可以用 \hskip 命令 (第 \hskip 页) 生成特定大小的水平粘连。命令 $\text{\hskip 6pt plus 2pt minus 3pt}$ 生成一个水平粘连，其自然尺寸为6点，伸长量为2点，收缩量为3点。类似地，你可以用 \vskip 命令 (第 \vskip 页) 生成特定大小的竖直粘连。

要理解什么是伸长量和收缩量，最好方法是看粘连的一个实际例子。假设你要用三个盒子和两个粘连项构造一个水平盒子项目，如下图：



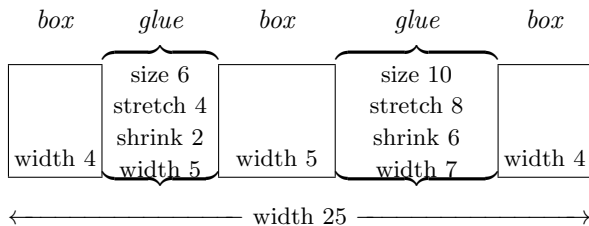
这里的度量单位可以是点，毫米，或其他任何单位。如果水平盒子可以呈现它的自然宽度，那么盒子内的每个粘连项也呈现它的自然宽度。从而水平盒子的总宽度就是它个各部分宽度之和，即29点。

接下来，假设这个水平盒子需要比 29 单位还宽，比方说 35 单位。这是可能出现的，比如水平盒子可能需要占据行宽为 35 单位的整行。由于 $\text{T}_{\text{E}}\text{X}$ 不可能改变盒子的宽度，它通过增加粘连项的宽度生成所需的额外间距。现在的图形看起来是这样的：



各粘连项增加的宽度并不相等；增加的宽度与它们的伸长量成比例。由于第二个粘连项的伸长量是第一个的两倍，它的宽度增加了四个单位，而后者只增加了两个单位。粘连允许任意伸长，尽管 $\text{T}_{\text{E}}\text{X}$ 有时不愿意超出定义中给出的伸长量。

最后，假设这个水平盒子需要比 29 单位还窄，比如说 25 单位。此时 $\text{T}_{\text{E}}\text{X}$ 将减少粘连项的宽度。图形看起来是这样的：



各粘连项减少的宽度于它们的收缩量成比例。第一个粘连项的宽度减少了一个单位，而第二个粘连项减少了三个单位。粘连收缩时不能超出定义中给出的收缩量，而伸长时却不受限制。从这个重要意义上说，收缩和伸长有不同的表现。

有一个不错的经验法则可用于设定粘连，即取它的自然尺寸为文档最美观时的间距大小，取它的伸长量为文档开始变难看之前 $\text{T}_{\text{E}}\text{X}$ 能添加的最大间距，取它的收缩量为文档开始变难看之前 $\text{T}_{\text{E}}\text{X}$ 能去掉的最大间距。你也许需要通过试验确定这些值。

以 `\fil`，`\filll`，或 `\fillll` 为单位设定粘连的伸长量，就可以生成可无限伸长的粘连。以 `\filll` 为单位的粘连相比以 `\fil` 为单位的粘连有更加无限的伸长能力，以 `\fillll` 为单位的粘连相比以 `\filll` 为单位的粘连有更加无限的伸长能力。你很少需要使用 `\fillll` 粘连。伸长量为 `2fil` 的粘连，其伸长能力是伸长量为 `1fil` 的粘连的两倍，对其他类型的可无限伸长粘连情形类似。

在 $\text{T}_{\text{E}}\text{X}$ 给各个粘连项分配额外间距时，如果存在可无限伸长的粘连，它们将分到全部间距。可无限伸长的粘连在设定文本左对齐，右对齐或者居中时特别有用：

- 要让文本左对齐，可以在它右边放上一个可无限伸长的水平粘连。这个粘连将占用该行所有可用的多余间距。你可以用 `\leftline` 命令（第 ``leftline'` 页）或 `\raggedright` 命令（第 ``raggedright'` 页）达到此目的。
- 要让文本右对齐，可以在它左边放上一个可无限伸长的水平粘连。如同上面，这个粘连将占用该行所有可用的多余间距。你可以用 `\rightline` 命令（第 ``rightline'` 页）达到此目的。
- 要设定居中文本，可以在它两边各放上一个同样的可无限伸长的水平粘连。这两个粘连项将平分该行的多余间距。你可以用 `\centerline` 命令（第 ``centerline'` 页）达到此目的。

类似地，你也可以指定可无限收缩的粘连。可无限收缩粘连可以作为负间距。注意 `fil` 等只可用于指定粘连的伸长量和收缩量——它们不能用于指定自然尺寸。

· `group`) 是文稿中 $\text{T}_{\text{E}}\text{X}$ 视为一个单元的一部分。要表示一个编组，你可以用括号 `{` 和 `}` (或者类别码合适的其他字符) 将它括起来。

编组的重要特性在于，当一个编组结束时，在其中所作的任何非全局的定义或赋值都消失了。假如你这样写：

```
Please don't pour {\it any} more tea into my hat.
```

`\it` 控制序列让 $\text{T}_{\text{E}}\text{X}$ 设定 ``any'` 为意大利体，但它不会影响其他文本。再举一个例子，如果你在编组中用 `\hsize` 命令（第 ``hsize'` 页）改变行宽，在 $\text{T}_{\text{E}}\text{X}$ 完成这个编组后，行宽将恢复为之前的取值。

编组也可以作为一种控制间距的方法。举个例子，如果你这样写：

```
\TeX for the Impatient and the Outpatient too.
```

你将会得到下列结果：

```
 $\text{T}_{\text{E}}\text{X}$ for the Impatient and the Outpatient too.
```

这是由于控制序列 `\TeX` (它生成 $\text{T}_{\text{E}}\text{X}$ 标识) 吸收了其后的空格。你所需要的也许是这样：

```
 $\text{T}_{\text{E}}\text{X}$  for the Impatient and the Outpatient too.
```

要得到这种结果，其中一种方法就是将 `\TeX` 括到一个编组里面：

```
{\TeX} for the Impatient and the Outpatient too.
```

其中的右花括号阻止了控制序列吸收空格。

。 `hbox`, horizontal box) 是 $\text{T}_\text{E}\text{X}$ 从左到右逐个放置水平列表的项目而构造出来的盒子。水平盒子, 作为一个单元, 既不是本质上水平的也不是本质上竖直的, 即它可以出现在水平列表或竖直列表中。你可以用 `\hbox` 命令 (第 `'hbox'` 页) 构造水平盒子。

。 `header`) 是放在每个页面顶部, 位于正文之上的素材。一个简单报告的页眉可能由左侧的标题和右侧的“第 n 页”组成。通常页眉只包含一行文本, 你可以通过给 `\headline` (第 `'headline'` 页) 指定一个记号列来设定页眉。在 plain $\text{T}_\text{E}\text{X}$ 中默认页眉是空白的。你也可以生成多行页眉; 请参阅第 `'bighead'` 页中介绍的方法。

。 `height`) 是盒子在基线之上的距离。

。 `horizontal list`) 是 $\text{T}_\text{E}\text{X}$ 位于某种水平模式时 (即组装段落或水平盒子时) 生成的一系列项目。见下面的“水平模式”。

。 $\text{T}_\text{E}\text{X}$ horizontal mode) 之一: 普通水平模式用于组装段落, 而受限水平模式用于组装水平盒子。只要 $\text{T}_\text{E}\text{X}$ 位于水平模式中, 它的胃 (见“ $\text{T}_\text{E}\text{X}$ ” depth height, 第 `'texpouxi'` 页) 就在构造项目 (盒子、粘连、惩罚等) 的水平列表。 $\text{T}_\text{E}\text{X}$ 逐个排版列表中的项目, 从左到右。

水平列表不能包含任何用本质上的竖直命令, 比如 `\vskip`, 生成的项目。

- 如果 $\text{T}_\text{E}\text{X}$ 正在普通水平模式中组装水平列表, 并碰到一个本质上的竖直命令, 它将结束当前段落并进入竖直模式。
- 如果 $\text{T}_\text{E}\text{X}$ 正在受限水平模式中组装水平列表, 并碰到一个本质上的竖直命令, 它将报错。

这两个命令你也许会认为是本质上水平的, 实际上却是本质上竖直的: `\halign` (第 `'halign'` 页) 和 `\hrule` (第 `'hrule'` 页)。在 *The $\text{T}_\text{E}\text{X}$ book* 第 286 页中列出了所有本质上的竖直命令。

对于受限水平模式, 你要知道这个微小但是重要的特性: 。在实践上, 这意味着在组装水平盒子时 $\text{T}_\text{E}\text{X}$ 无法处理类似段落的文本, 即需要断行的文本。在水平盒子内部, 通过把类似段落的文本放入一个竖直盒子中, 你就可以绕过此限制。其他类似方法也可行, 比如将一个水平阵列放在水平盒子内部。

· $\text{T}_{\text{E}}\text{X}$ hyphenation)。 $\text{T}_{\text{E}}\text{X}$ 并不急于插入连字符，它更喜欢通过调整单词间隔找到合适断行点，从而将单词用一行移动到另一行。 $\text{T}_{\text{E}}\text{X}$ 足够聪明，它可以理解已经出现在单词中的连字符。

你可以用如下几种方式控制 $\text{T}_{\text{E}}\text{X}$ 的连字：

- 你可以用 `\-` 命令（第 ``@minus'` 页）插入自定连字符，以允许 $\text{T}_{\text{E}}\text{X}$ 在特定位置连字化。
- 你可以用 `\hyphenation` 命令（第 ``hyphenation'` 页）告诉 $\text{T}_{\text{E}}\text{X}$ 在整个文档中如何对某个单词连字化。
- 你可以将单词放入一个水平盒子中，以阻止 $\text{T}_{\text{E}}\text{X}$ 将它连字化。
- 你可以设定一些惩罚值，比如 `\hyphenpenalty`（第 ``hyphenpenalty'` 页）。■

如果单词中包含显式或自定连字符， $\text{T}_{\text{E}}\text{X}$

· ‘

· insertion) 是一个竖直列表，它包含即将在页面建造完成后插入到该页面的素材。¹¹ 脚注和图片就是插入项的例子。在 plain $\text{T}_{\text{E}}\text{X}$ `\footnote\topinsert\midinsert \pageinsert \insert ‘footnote’-‘endofinsert’` $\text{T}_{\text{E}}\text{X}$ *The $\text{T}_{\text{E}}\text{X}$ book* 122–125

· interline glue) 是 $\text{T}_{\text{E}}\text{X}$ 在竖直列表的每个盒子（第一个盒子除外）之前插入的粘连。通常指定行间粘连以让各盒子基线间保持相同的距离。行间粘连的值由 `\baselineskip`、`\lineskip` 和 `\lineskiplimit` 这三个参数合起来确定（第 ``baselineskip'` 页）。

· item) 这个术语常用来表示水平，竖直或数学列表（即 $\text{T}_{\text{E}}\text{X}$ 在水平，竖直或数学模式中建造的项目列表）的一个组成元素。

· justified text) 是两边对齐排版的文本。而非对齐文本是单边或两边“不对齐”排版的文本。老式打印机排版的文档几乎总是左对齐的。虽然 $\text{T}_{\text{E}}\text{X}$ 默认生成两边对齐的文档，如果你需要也可以生成左对齐或右对齐的文档（或一些文本行）。你也可以让 $\text{T}_{\text{E}}\text{X}$ 将一些文本行居中，从而让两边都不对齐。用 `\leftskip`、`\rightskip` 和 `\raggedright` 命令（第 ``leftskip'`、``raggedright'` 页）可以达到这些目的。

在生成对齐文本时， $\text{T}_{\text{E}}\text{X}$ $\text{T}_{\text{E}}\text{X}$

¹¹ $\text{T}_{\text{E}}\text{X}$ `\insert ‘insert’` $\text{T}_{\text{E}}\text{X}$ $\text{T}_{\text{E}}\text{X}$ $\text{T}_{\text{E}}\text{X}$

· kern) 表示对竖直或水平列中的项目间的距离作一定调整。紧排可正可负。将正紧排放入两个项目间, 将以该紧排的大小拉开两个项目。而将负紧排放入两个项目间, 将以该紧排的大小拉近两个项目。例如, 下列文本:

```
11\quad 1\kern1pt 1\quad 1\kern-.75pt 1
```

将生成类似下面的几个字符对:

```
11 11 11
```

在竖直模式中可以用紧排改变特定两行的间距。

大小为 d 的紧排与大小为 d 且无伸缩的粘连非常相似。紧排和粘连两者都在相邻项目间插入或删除一定间距。本质区别在于, 若两盒子间只有紧排, \TeX ¹²

· leaders), 你可以用某个模式的复本填充空间, 比如在目录中的用重复的点填充标题和页码之间的空间。指引体 (leader) 是模式的单个复本。要指定一个指引线, 你需要给出三方面的信息:

- 1) 单个指引体是怎样的
- 2) 要填充的空间有多大
- 3) 模式的复本在空间内应该如何组织

\TeX 有三个用于指定指引线的命令: `\leaders`, `\cleaders` 和 `\xleaders` (第 `'leaders'` 页)。各命令的参量指定了指引体。命令之后必须是一个粘连; 粘连的大小指定了要填充空间的大小。命令的选择决定了指引体在空间内如何组织。

这里有个例子展示了 `\leaders` 的工作方式:

```
\def\dotting{\leaders\hbox to 1em{\hfil.\hfil}\hfil}
\line{The Political Process\dotting 18}
\line{Bail Bonds\dotting 26}
```

其中我们将指引线及相关粘连放在一个宏定义中, 以方便我们在两个地方使用它。这个例子的输出结果是:

```
The Political Process . . . . . 18
Bail Bonds . . . . . 26
```

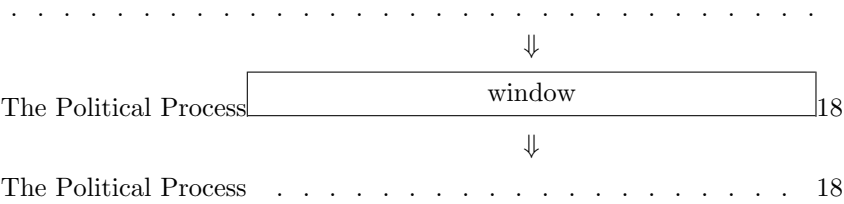
¹² kern”翻译为“紧排”正由于这个原因, “紧”应该理解为“紧固”。 \TeX

\TeX 10 ‘To’ 的紧排, 将字母 ‘o’ 的左边缘拉到字母 ‘T’ 下边。没有这个紧排, 你将得到 “Top” 而不是 “Top” ——两者的差别是微小的但却是明显的。 \TeX 用某个字体排版文本时自动插入的紧排的位置和大小, 是在该字体的度量文件 (`.tfm` 文件) `depth height plus 0.1em minus 0.1em` 中指定的。

`\leaders` 之后的水平盒子指定了指引体，即包含居中点的 1em 宽的水平盒子。空间将用这个盒子的复本填充，实际上是用中心相隔 1em 的点填充。随后的 `\hfil`（宏定义末尾那个）是一个粘连，它给出所要填充空间的大小。在这个例子中，它就是填满文本行所需的空
间。我们选择 `\leaders` 而不是 `\cleaders` 或 `\xleaders`，是为了让不同
的点相互对齐。

一般地，所要填充的空间就作为包含指引体的多个复本的窗体。
`TEX` 插入尽可能多的复本，但通常会有些空间剩下——或者是由于
指引体在窗体内的位置问题，或者是由于窗体的宽度不是指引体宽度
的整数倍。这三个命令的差异在于，在窗体内如何安排指引体和分
配剩余空间：

- 对于 `\leaders`，`TEX` 首先生成一行指引体复本。然后该行复本的起
始位置与包含 `\leaders` 命令的最内侧盒子 *B* 的左侧对齐。在上面的
两行例子中，*B* 就是由 `\line` 生成的盒子。整个落入窗体的指引
体将被放入 *B* 中，而左右两边剩下的空间将空着。图形如下：



这个过程确保在上面的两行例子中，两行中的各个点是竖直对齐的
(由于用 `\line` 生成的水平盒子的基准点是竖直对齐的)。

- 对于 `\cleaders`，`TEX` 平分窗体两边的剩余空间以让指引线在窗体
中居中。剩余空间总是小于单个指引体的宽度。
- 对于 `\xleaders`，`TEX` 将剩余空间在窗体内均匀分布。也就是说，
如果剩余空间是 *w*，指引体被重复 *n* 次，`TEX` 将在相邻指引体中间
及指引线两端填充宽度为 $w/(n + 1)$ 的空间。其效果通常是将指引
体散开一点。和 `\cleaders` 类似，`\xleaders` 的剩余空间也总是小
于单个指引体的宽度。

到目前为止，我们都假定指引线由横向排列的水平盒子组成的。实
际上指引线还有两个可能的变种：

- 1) 你可以用标线代替水平盒子作为指引体。`TEX` 将让标线尽可能宽以
让它填满该粘连（从而这三个命令是等价的）。
- 2) 你可以不在水平列表而在竖直列表中使用它，以在页面中生成向下
排列的竖直指引线。在这种情形中指引线后面需要给出竖直粘连。

请参阅 *The T_EXbook* 第 223--225 页，其中给出了 T_EX 用于排版指引线的精确规则。

. ligature) 是排版的文档中替换特定的相邻字符串的单个字符。例如，在高质量的排版系统中，单词 ``office'` 将被排版为 ```office''`，而不是 ```office''`。连写信息已经嵌入在你使用的字体中，因此什么也不用做你就可以让 T_EX 生成它们。(通过将输入写成 ``off{f}ice'` 可以取消 ```office''` 中的连写，正如我们刚才所做的。) 利用此连写机制，对单词的开头或者末尾字母，T_EX `\noboundary` `'noboundary'`

T_EX `'Æ'`，可以用命令得到（见“欧洲语言字母和连写”，第 `fornlets'` 页）。■

. line break) 是文档中 T_EX 结束一行并开始新行的位置。在处理你的文档时，T_EX T_EX “” 值给各种形态的丑陋断行点——单词间隔太大或太小的行，连续以连字符结尾的行，诸如此类。然后它选择让总缺陷值最小的断行点。请参阅 *The T_EXbook* 第 96--101 页中的 T_EX 的断行规则的完整说明。

你可以用下面几种方式控制 T_EX 对断行点的选择：

- 你可以在 T_EX 形成段落时构建的水平列表的某处插入惩罚项（第 `hpenalty'` 页）。正惩罚值阻碍 T_EX 在该处断行，而负惩罚值——即奖励值——鼓励 T_EX 在该处断行。大于或等于 10000 的惩罚只阻止断行，而小于或等于 -10000 强制断行。用 `\break` 和 `\nobreak` 命令（第 `hbreak'`，`hnobreak'` 页）可以达到相同效果。
- 你可以用 `\-` 命令（第 `@minus'` 页）插入自定连字符，以允许 T_EX 在特定位置连字化。你也可以控制 T_EX 在你的文档中如何连字化（见 “” depth height，第 `'` 页）。
- 你可以在两个单词间用 `\slash`（第 `slash'` 页）插入斜线符号 (/)，并允许 T_EX 在斜线后断行，比如 ``furlongs\slash fortnight'`。■
- 你可以通过在两个特定单词间插入带子 (~)，以禁止 T_EX 在该处断行。
- 你可以给 T_EX 的断行参数赋予不同的值，以调整断行时附加的惩罚值。
- 你可以将一个单词或一系列单词放在一个水平盒子中，从而阻止 T_EX 盒子里面的任何位置断行。

知道在什么地方 $\text{T}_{\text{E}}\text{X}$ 可以断行也是有益的：

- 在粘连处，只要：
 - 1) 粘连之前是下列项目之一：盒子、自定项目（比如自定连字符）、公式结尾、小玩意或者用 `\mark` 或 `\vadjust` 或 `\insert` 生成的竖直素材
 - 2) 粘连不是数学公式的一部分

在粘连处断行时， $\text{T}_{\text{E}}\text{X}$ 让断行点位于粘连的左边缘，并忽略粘连的其他部分。

- 在紧排处，只要其后直接跟着粘连，且该紧排不在数学公式中
- 在数学公式结尾处，只要其后直接跟着粘连
- 在惩罚项处，即使该惩罚项在数学公式中
- 在自定可断点处

在断行时， $\text{T}_{\text{E}}\text{X}$

。 `list`）是一列用于组成水平列表、竖直列表或数学公式的项目（盒子、粘连、紧排等）。请参阅 “” `depth height`（第 ‘1’ 页），“” `depth height`（第 ‘1’ 页）。

。 ‘

。 `macro`）是一个定义，它给出了 $\text{T}_{\text{E}}\text{X}$ 输入文本的一种模式的名称。¹³ 这个名称可以是一个控制序列或一个活动字符。其中的模式称为“替换文本”。用于定义宏的原始命令是 `\def` 控制序列。

举个简单例子，假设你有一个文档，其中多次出现 ‘ $\cos \theta + i \sin \theta$ ’ `depth height plus 0.1em minus 0.1 em` 这串字符。为避免每次都写一遍，你可以给它定义一个宏：

```
\def\arctheta{\cos \theta + i \sin \theta}
```

这样每次需要这串字符时，用 ‘`\arctheta`’ `depth height plus 0.1em minus 0.1 em` 命令“调用”这个宏就可以得到它。比如，‘ e^{\arctheta} ’ `depth height plus 0.1em minus 0.1 em` 将给出 ‘ $e^{\cos \theta + i \sin \theta}$ ’ `depth height`。

但宏的真正威力在于它可以带有参数。在调用一个带有参数的宏时，你给出用于替换这些参数的参量。例如，假设你这样写：


```
\def\arc#1{\cos #1 + i \sin #1}
```

记号 #1 表示宏的第一个参数，在这个例子中宏只有一个参数。现在你可以用 ‘`\arc {2t}`’ depth height plus 0.1em minus 0.1 em 得到 ‘ $\cos 2t + i \sin 2t$ ’ depth height。

更一般地，一个宏最多可以有九个参数，在宏定义中它们分别用 ‘#1’, ‘#2’ 等表示。T_EX 提供两种类型的宏参数：定界参数和非定界参数。简单来说，定界参数对应一个由特定记号序列（即定界子）定界（即结束）的参量，而非定界参数对应一个无需定界子结束的参量。我们先解释只有非定界参数时宏如何运作，然后解释包含定界参数时宏如何运作。

当宏只有非定界参数时，这些参数必须依次出现在宏定义中，。对这种宏的调用由该宏的名称，以及后面跟随的各个参量组成。每个参量对应一个参数，参量可以是

- 不为左或右花括号的单个记号，或者
- 一个包含在配对的左右花括号之间的记号序列。¹⁴

T_EX 在食道中展开它所遇到的宏（见 “T_EX” depth height，第 ‘`\texpouxi`’ 页）， 方法是将替换文本中的每个参数替换为它所对应的参量。宏展开后所得到的文本中可能还包含其他宏。当 T_EX 遇到这种嵌套的宏调用时，它将立即展开此宏调用，完成后再查看后续内容。¹⁵ 当 T_EX 的食道得到无法再展开的原始命令时，T_EX 就将该命令送入 T_EX 的胃。宏展开的先后顺序有时候是至关重要的，为帮助你理解这个问题，下面我们将给出一个例子。

假设你给 T_EX 提供如下输入：

```
\def\aa#1#2{\b#2#1\kern 2pt #1}
\def\b{bb}
\def\c{\char49 cc}
\def\d{dd}
\ac{e\d} % Call on \a.
```

则与 #1 对应的参量为 \c，与 #2 对应的参量为 e\d。T_EX 按照如下步骤展开宏调用：

```
\b e\d\c\kern 2pt \c
bbe\d\c\kern 2pt \c
\d\c\kern 2pt \c  (‘b’, ‘b’, ‘e’ 送到胃里) dd\c\kern 2pt
\c \c\kern 2pt \c  (‘d’, ‘d’ 送到胃里) \char49 cc\kern 2pt \c
\c  (‘\char’, ‘4’, ‘9’, ‘c’, ‘c’, ‘\kern’, ‘2’, ‘p’, ‘t’ 送到胃里)
\char49 cc (‘\char49’, ‘c’, ‘c’ 送到胃里)
```

¹⁴

¹⁵ `\expandafter`

‘b’, ‘c’, ‘d’ ‘e’, 以及控制序列 ‘\kern’ 和 ‘\char’ 都是无法再展开的原始命令。

宏也可以包含“定界参数”，它们可以和非定界参数混合使用。定界参数的想法是，T_EX 通过寻找标记参量结束的特定记号序列（即定界子）确定对应的参量。也就是说，在查找这种参量时，T_EX 选取从当前位置到定界子之间的所有记号，不包含该定界子。

要指定一个定界参数，你可以在 ‘#n’（n 必须在 1 和 9 之间）后面加上一个或多个充当定界子的记号。定界子一直延续到下一个 ‘#’ 或 ‘{’ 为止——这是有道理的，因为 ‘#’ 开始另一个参数，而 ‘{’ 开始替换文本。

定界子不能是 ‘#’ 或者 ‘{’，因而你可以从后面的字符辨别出定界参数和非定界参数。

如果参数之后的字符为 ‘#’ 或 ‘{’，你得到一个非定界参数，否则你得到一个定界参数。注意这两种参数的参量形式不同——非定界参数匹配单个记号或者围在花括号中的记号序列，而定界参数匹配任意多个记号，即便是零个也可以。

下面的例子用到了两个定界参数：

```
\def\diet#1 #2.{On #1 we eat #2!}
```

其中第一个参数用单个空格定界，而第二个参数用英文句号定界。如果你这样写：

```
\diet Tuesday turnips.
```

你将得到文本 “On Tuesday we eat turnips!”。但若将定界记号放在编组中，T_EX

```
\diet {Sunday mornings} pancakes.
```

得到 “Sunday morning Sunday mornings pancakes.”。当你将空格符作为定界子时，行尾符一般同样定界了对应的参量；这是因为 T_EX 在宏机制运作之前就将行尾符都转换为空格了。

偶尔在你所定义的宏里面需要将 ‘#’ 用做一个有意义的字符。当你在宏定义中实际上又定义了第二个宏时，你多半需要这样做。怎样处理第二个宏的参数，以让 T_EX 不会混淆它们俩呢？答案是，在第一个宏展开后需要一个 ‘#’ depth height plus 0.1em minus 0.1 em 的地方写上两个 ‘#’ depth height。假如你写出下面的宏定义：

```
\def\first#1{\def\second##1{#1/#1}}
```

则对 ‘`\first{One}`’ depth height plus 0.1em minus 0.1 em 的调用将把 ‘`\second`’ depth height plus 0.1em minus 0.1 em 定义为：

```
\def\second#1{One/#1}
```

而随后的调用 ‘`\second{Two}`’ depth height plus 0.1em minus 0.1 em 将得到文本 ‘One/Two’。

还有一些命令提供了宏的其他定义方式（见第 ‘`\mac1`’--‘`\mac2`’ 页）。在 *The T_EXbook* 第 20 章中有关于宏的完整规则。

· T_EX magnification) 因子 $f/1000$ ，其中 f 是 `\mag` 参数（第 ‘`\mag`’ 页）的值。由于 `\mag` 的默认值为 1000，在一般情形下文档是照常排版的。若以后需要缩印文档，增加文档的放大率通常就比较有用。

你也可以缩放一个字体，以得到比它的“设计大小” depth height plus 0.1em minus 0.1 em 更小或更大的字体。对每个使用的缩放字体，你需要提供它的字形文件给设备驱动程序（见 “” depth height，第 ‘’ 页）——除非该字体已经内嵌在打印机中，而设备驱动程序也知道它。在用 `\font` 命令（第 ‘`\font`’ 页）定义一个字体时，你可以用单词 ‘`scaled`’ depth height plus 0.1em minus 0.1 em 指定放大率。比如：

```
\font\largerbold = cmbx10 scaled 2000
```

将 ‘`\largerbold`’ 定义为两倍大小的 `cmbx10` 字体（计算机现代粗体扩展的10点字体），它的各个字符形状也统一地放大了两倍。

为方便起见，很多计算机中心提供了按1.2比例放大的字体，这些字体对应的放大率为 1200，1440，等等。在 T_EX 中这些放大率有特别的名称：‘`\magstep1`’ 表示 1200，‘`\magstep2`’ 表示 1440，依此类推直到 ‘`\magstep5`’。特殊值 ‘`\magstephalf`’ 对应 $\sqrt{1.2}$ 倍的放大率，它看上去位于 ‘`\magstep0`’（不放大）和 ‘`\magstep1`’ 的正中间。例如：

```
\font\bigbold = cmbx10 scaled \magstephalf
```

若想指定一个最终文档中的尺寸，让它不受放大率影响，你可以在单位前面加上 ‘`true`’。例如，‘`\kern 8 true pt`’ 将生成 8 点大小的紧排，不管放大率为多少。

· margin) 确定了一个矩形边框，该边框通常包含页面的印刷内容。你也可以让 T_EX 在这个边框之外印刷内容，但你必须用某种明确方法将内容移动到哪里。T_EX 将页眉和页脚放在边框外面。

这个矩形边框通过它的左上角位置，宽度，以及深度来定义。左上角位置用 `\hoffset` 和 `\voffset`（第 ‘`\voffset`’ 页）参数来定义。默认将这个位置放在离页面顶端和左端各一英寸的地方，这对应于 `\hoffset`

和 `\voffset` 都取值为零。¹⁶ 边框的宽度用 `\hsize` 给出，而深度用 `\vsize` 给出。

这些规定给出的结果是：

- 左边距等于 `\hoffset + 1in`。
- 右边距等于页面宽度减去 `\hoffset + 1in + \hsize`。
- 上边距等于 `\voffset + 1in`。
- 下边距等于页面长度减去 `\voffset + 1in + \vsize`。

从这些信息中你可以知道，要改变页面边距应该修改哪些参数。

对 `\hoffset`、`\voffset` 或者 `\vsize` 的修改在 \TeX 开始新页面时才生效。也就是说，如果你在页面中间改变它们，这些改动将只影响页面。然而，对 `\hsize` 的修改却会立即生效。

`\mark`) 是可以插入水平、竖直或者数学列表中，并在输出例行程序中还原的项目。标记可用于记录即将出现在页眉中的内容等。每个标记带有一列记号——即“标记文本” `depth height`。`\mark` 命令（第 `\mark` 页）以这样的记号列作为参量，并将包含这个记号列（在展开之后）的项目添加到 \TeX 当前构造的列表中。`\topmark`、`\firstmark` 和 `\botmark` 命令（第 `\topmark` 页）用于取回页面上的各种标记。这些命令经常在页眉和页脚中用到。

这里有个简化的例子。假设你定义了节标题宏如下：

```
\def\section#1{\medskip{\bf#1}\smallskip\mark{#1}}
% #1 is the name of the section
```

调用这个宏将生成粗体的节标题，并将标题名记录为一个标记。现在你可以对每个打印页定义页眉如下：

```
\headline = {\ifodd\pageno \hfil\botmark\quad\folio
\else \folio\quad\firstmark\hfil \fi}
```

这样，每个偶数页（左手页）页眉是页码加上该页第一个标题名，而每个奇数页（右手页）页眉是该页最后一个标题名加上页码。特殊情形，比如该页没有节标题，通常也会按照 `\firstmark` 和 `\botmark` 的规定正常显示。

在用 `\vsplit` 命令（第 `\vsplit` 页）分割页面时，用 `\splitfirstmark` 和 `\splitbotmark` 命令（第 `\splitfirstmark` 页）就可以取得分割的这部分中的第一个和最后一个标记的标记文本。

在 *The \TeX book* 第 258–260 页中，对如何生成和取回标记作了更准确的解释。

¹⁶ `\hoffset` `\voffset` (0,0)

. math mode) 是 \TeX 构造数学公式时所在的模式。 \TeX 有两种数学模式：文内数学模式用于构造放在文本行中的公式，而陈列数学模式用于构造单独一行显示的公式。括在 $\$$ 里面的公式表示一个文内数学公式，而括在 $\$$ 里面的公式表示一个陈列数学公式。这两种数学模式有个重要特性，即。 *The \TeX book* 第 290–293 页详细介绍了在数学模式中 \TeX 如何处理各种不同命令。

. mathcode) 是一个数, \TeX 256 \TeX
 \TeX 0 15 \TeX *The \TeX book* 154 =3 \TeX
 \TeX 11 12 *t* \TeX

1)

2) f

3) s

4) fsF .

5) n

6) Fnc

7) tc

8) c

(3) 和 (7) 中的上下文相关性意味着， \TeX 在排版数学字符时，要先看完该字符所在的整个公式。比如对于公式 ‘ $\$a\over b\$$ ’ depth height , \TeX $a\over$

4096 c + 256 f + n c f n ASCII \mathcode ‘mathcode’ \TeX 1112
 \TeX \mathcode
7 \TeX \fam ‘fam’ 0 \fam = 0 \fam 0 15 \TeX 0 0 7
 \TeX \fam -1

. mathematical unit), 记为 ‘mu’ depth height , 是一个用于设定数学公式中的粘连大小的距离单位。见 “” depth height (第 ‘ 页)。

. \TeX \TeX ‘texpouxi’ mode) 的其中一种:

- 普通水平模式 (组装段落)
- 受限水平模式 (组装水平盒子)
- 普通竖直模式 (组装页面)
- 内部竖直模式 (组装竖直盒子)
- 文内数学模式 (组装在文本中显示的公式)
- 陈列数学模式 (组装单独一行显示的公式)

模式描述了 \TeX 正在组装的实体的类型。

由于你可以将某种实体嵌入另一种实体中，比如将一个竖直盒子放入数学公式中， \TeX 记录的不只是一个模式，而是一整列模式（即计算机科学中所说的“栈”）。假设 \TeX 在模式 M 中，然后遇到某些东西而进入新模式 M' 。当它完成在模式 M' 的工作时，它将继续在模式 M 中的工作。

\mu (muglue) 是一种仅用于数学公式中的粘连，它的度量单位是 \mu (数学单位)。1 \mu 的大小与 $1/18 \text{ em}$ 相等，而 1 em 的大小取自第2族数学字体。 \TeX 根据上下文自动调整 \mu 的大小。比如同样是 2 \mu 大小的粘连，在下标中一般就比在普通文本中小一些。要生成数学粘连，你必须使用 \mskip 命令。例如， $\text{\mskip 4\mu plus 5\mu}$ depth height plus 0.1em minus 0.1 em 生成了一个自然大小为 4 \mu ，伸长量为 5 \mu 的数学粘连。

\TeX number) 是指正整数或负整数。 \TeX

- 1) 52
 - 2) '14
 - 3) "FF0
 - 4) ASCII ' ' '\
- '+' '-'

0-7 0-9 A-F 0 15 The \TeX book 43-44

'A' 和 'F' 之间的字母却不能结束一个十六进制数。你也可以用一个或多个空格结束一个数， \TeX 通常将忽略这些空格。¹⁷

上述的第四种形式用字符的ASCII编码表示一个数。 \TeX 同样忽略此形式的数后面的空格。一个数按这种形式可以写成 'c 或者 '\c 。第二种写法虽然长一点，但优点在于可以用于字符，甚至是 '\ ， '\% 或者 '\^M 。不过它确实也有一个技术上的缺点：比如当 \TeX 展开 \edef 或 \write 命令的记号序列时，出现在数中间的能展开的 '\c 将同样被展开；而这多半不是你想要的。

下面是十进制数78的全部有效的表示方式：

78 +078 "4E '116 'N '\N

你不能在文本中单独使用一个数，因为它不是命令。但是，将它放在 \number 命令（第 \number 页）之后，你就可以在文本中插入这个数的十进制形式；或者将它放在 \romannumeral 命令之后，以得到它的罗马数字形式。

¹⁷ \TeX

你也可以用小数，即带有小数部分的数，来指定一个尺寸（见“” depth height，第`'页）。小数包含一个可以在最前面的小数点。小数点的圆点也可以用逗号代替。小数之前也可以加上正号或负号。因此`.5in`、`-3.22pt`和`+1,5\baselineskip`都是有效的尺寸。但是，除了作为尺寸的“因子”，即乘数，你不能在地方使用小数。

. ordinary mode) 是 T_EX 分段为行或组行为页时所在的模式。见“” depth height (第`'页)， “” depth height (第`'页)。

. outer) 宏是在某些 T_EX 高速处理记号的环境中无法使用的宏。将一个命令定义为外部的，目的是使 T_EX 能够尽早捕获到错误。在定义宏时，你可以用 `\outer` 命令 (第`outer' 页) 让它成为外部的。

在下列这些环境中，你不可以使用外部宏：

- 在命令的参量中
- 在宏定义的参数文本或替换文本中
- 在阵列的导言中
- 在条件测试的跳过部分中

外部环境是你可以使用外部宏的环境，即刚才列出的环境除外的环境。

例如，下面的输入就是外部宏的禁止用法：

```
\leftline{\proclaim Assertion 2. That which is not inner
is outer.}
```

`\proclaim` 宏 (第`@proclaim' 页) 在 plain T_EX 中被定义为外部的，但这里却将它放在 `\leftline` 的宏参量中。

. T_EX \box255 T_EX output routine) 以加工素材并最终将它送到 .dvi 文件中。输出例行程序可以执行进一步的加工，比如插入页眉，页脚和脚注。Plain T_EX 提供了一个默认的输出例行程序，它在每个页面底部插入居中的页码。通过提供不同的输出例行程序，你可以达到诸如双栏输出的效果。你可以认为输出例行程序只有一个职责：以某种方式处理掉放在 `\box255` 中的素材。

当前的输出例行程序由 `\output` (第`output' 页) 的值定义，它是一个记号列。当 T_EX 准备好生成页面时，它就展开该记号列。

实际上不用修改 plain T_EX 的输出例行程序，你就能对它的行为做出一些简单改动。例如，通过赋予一个记号列给 `\headline` 或 `\footline` (第`footline' 页)，你就可以让 T_EX 生成和通常情形不同的页眉或页脚。

输出例行程序也负责收集任何插入项；它将这些插入项和任何“装饰”（比如页眉页脚）合并到页面主要内容中，并将所有这些素材包装

在一个盒子里；最后它用 `\shipout` 命令（第 `'\shipout'` 页）将这个盒子送到 `.dvi` 文件中。虽然输出例行程序大多是这样运行的，但有特定用途的输出例行程序会有不同表现。

·

· `TEX` page)，并将它们送往输入例行程序。在处理文档时，`TEX`

`\eject 'eject' TEX`

The T_EXbook 111

`\box255 \box255 \box255 \shipout .dvi TEX \box255`

`TEX TEX The TEXbook 110–114 TEX`

· page break) 是文档中 `TEX` 结束一个页面并（除非在文档结尾处）开始新页面的位置。在 “” depth height（第 `'` 页）中介绍了 `TEX` 选择分页点的过程。

你可以用以下几种方式控制 `TEX` 对分页点的选择：

- 你可以插入惩罚项（第 `'\vpenalty'` 页）到主竖直列的两个项目之间。正惩罚值阻碍 `TEX` 在此处分页，而负惩罚值——换言之，奖励值——鼓励 `TEX` 在此处分页。大于或等于 10000 的惩罚值阻止分页，而小于或等于 -10000 的惩罚值强制分页。你可以用 `\break` 和 `\nobreak`（第 `'\vbreak'` 页）命令得到相同的结果。
- 你可以通过改变 `TEX` 的分页参数的值来调整分页惩罚值。
- 你可以将主竖直列的一系列段落或者其他项目放入竖直盒子，从而阻止 `TEX` 在它们之间的任何位置分页。

一旦 `TEX` 选定了分页点，它就将该主竖直列中位于分页点之前的内容放入 `\box255`。接着它调用当前的输出例行程序处理 `\box255`，最终输出页面内容到 `.dvi` 文件。输出例行程序还得处理 `TEX` 加工页面时积累的插入项，比如脚注。

最好能知道 `TEX` 在什么地方可以分页：

- 在粘连处，只要在粘连之前的项目是一个盒子，小玩意，标记，或者插入项。当 `TEX` 在粘连处分页时，它把分页点放在该粘连间距的顶部，并忽略该粘连的其他值。
- 在紧排处，只要该紧排之后是一个粘连。
- 在惩罚处，这可能出现在段落的两行之间。

`TEX` 分页时将丢弃分页点之后的任意多个粘连，紧排和惩罚项。

· page layout): 页面大小, 四个方向的边距, 页眉和页脚, 若存在的话, 还有正文和页眉页脚之间的间隔大小。对上述这些 $\text{T}_{\text{E}}\text{X}$ 都有默认值: 它设定页面大小为 $8\frac{1}{2}$ 英寸乘 11 英寸, 四个边距约为一英寸, 没有页眉, 页脚是居中的页码。

边距由四个参数 `\hoffset`, `\voffset`, `\hsize` 和 `\vsize` 共同确定 (在第 `'` 页的“边距”概念中有如何调整它们的指南)。页眉是在每个页面顶部的一行内容, 位于上边距区域中。通过将一个记号列表赋值给 `\headline` 参数 (第 `'headline'` 页), 你就可以设定页眉。类似地, 页脚是在每个页面底部的一行内容, 位于下边距区域中。通过将一个记号列表赋值给 `\footline` 参数 (第 `'footline'` 页), 你就可以设定页脚。例如, 下面的输入:

```
\headline = {Baby's First Document\dotfill Page\folio}
\footline = {\hfil}
```

将给每个页面生成类似下面的页眉行

Baby's First Document.....Page 19

而页脚行是空白的。

利用标记, 你可以将当前主题或节标题放入页眉或页脚中。在“” depth height (第 `'` 页) 这里将详细解释如何做到这些。

· paragraph) 就是一系列输入行, 并用空行或者 `\par` 命令 (第 `'@par'` 页), 或者类似 `\vskip` 的本质上的竖直模式命令结束。更准确地说, 段落就是 $\text{T}_{\text{E}}\text{X}$ 在普通水平模式中处理的一系列命令。 $\text{T}_{\text{E}}\text{X}$ 收集整个段落的内容后, 就通过选择断行点 (见“” depth height, 第 `'` 页) 将它分为多行。得到的结果是一系列水平盒子, 以及它们之间的粘连, 行间惩罚和穿插的竖直素材。其中的每个水平盒子就是一个文本行, 而这些粘连就是行间粘连。

当 $\text{T}_{\text{E}}\text{X}$ 在竖直模式中碰到本质上为水平模式的命令, 它开始一个新段落。特别地, 在刚结束一个段落的时候, 它位于竖直模式, 因此在空白输入行之后的水平素材自然就开始下一个段落。本质上为水平模式的命令有好多, 但最常见的就是普通字符, 比如一个字母。

`\indent` 和 `\noindent` 命令 (见 `'indent'`, `'noindent'` 页) 也属于本质上为水平模式的命令, 它们让 $\text{T}_{\text{E}}\text{X}$ 在段落开始缩进或不缩进。其他水平模式命令出现在竖直模式时都会导致 $\text{T}_{\text{E}}\text{X}$ 隐性执行 `\indent`。一旦 $\text{T}_{\text{E}}\text{X}$ 开始一个新段落, 它就处于普通水平模式中。它首先执行 `\everypar` 中的命令, 然后着手收集该段落的项目, 直到它得到段

落结束的信号。在段落结尾处，它重置段落形状参数 `\parshape`，`\hangindent` 和 `\looseness`。

\TeX 一般将一个空行转换为一个 `\par` 命令。当它在水平模式碰到一个本质上的竖直模式命令时，它同样插入 `\par` 命令到输入中。因此结束段落的最终都是一个 `\par` 命令。

当 \TeX 收到一个 `\par` 命令时，它首先填满¹⁸ 当前段落。接着将该段落分为多行，并将所得结果添加到所在的竖列中，然后执行页面构建器（仅当所在的竖列就是主竖列时）。如果该段落是用本质上为竖直模式的命令结束的， \TeX 接下来将执行该命令。

parameter) 这个术语有两个不同的含义——它可以表示 \TeX 参数或者表示宏参数。

\TeX 参数是一个控制序列，它是某个值的名称。参数的值可以是数值，长度，一定大小的粘连或数学粘连，或者是一个记号列表。例如，`\parindent` 参数指定了 \TeX 在缩进段落开头所跳过距离的大小。

你可以对 \TeX 参数的控制序列取值或赋值。如果该控制序列出现在需要取值的地方， \TeX

```
\hskip\parindent
\parindent
\parindent = 2pc % (or \parindent 2pc)
\parindent
\parindent = 1.5\parindent
\parindent \parindent 1.5
 $\text{\TeX}$  The  $\text{\TeX}$ book 272–275  $\text{\TeX}$ 
 $\text{\TeX}$ 
 $\text{\TeX}$ 
\penalty ‘hpenalty’, ‘vpenalty’ 10000 –10000
```

plain \TeX . Plain \TeX The \TeX book \TeX Plain \TeX \TeX plain \TeX
 \TeX

Plain \TeX The \TeX book B 中给出了这些附加定义。它们应该同样包含在你的计算机系统的某个 `plain.tex` 文件中。

18

`\unskip \penalty10000 \hskip\parfillskip`

. primitive) 命令是在 \TeX 程序内部就定义好的命令。作为对比, 非原始命令或者是从宏定义得到的, 或者是从 \TeX 编写的其他形式的定义得到的。Plain \TeX 的命令由原始命令加上用原始命令定义的其他命令组成。

. reference point) 是盒子左边缘与基线的交点。在处理水平列表或竖直列表时, \TeX 用列表中各盒子的基准点水平或竖直对齐这些盒子 (见 “” depth height, 第 1 页)。

. register) 是用于存储数值的命名位置。它很像编程语言中的一个变量。 \TeX

```
box
count
dimen
muskip
skip
toks
```

```
0 255 n c '\cn' \muskip192 \count12 \skip0
```

```
\setbox3 = \hbox{lagomorphs are not mesomorphs}
\count255 = -1
```

```
3 '\box3' 19 -1 255
```

```
\TeX \pageno
```

```
Plain \TeX \TeX \newbox \newcount \newdimen \newmuskip \newskip
```

```
\newtoks '@newbox'
```

```
\TeX \TeX
```

```
\TeX \count0 \count9 \box255 \dimen0-\dimen9 \skip0-\skip9 \muskip0-
```

```
\muskip9 \box0-\box9 \box255 255 plain \TeX \count255 The
\TeXbook 122 346
```

```
\TeX \showthe 'showthe' '\showthe\dimen0'.
```

. restricted mode) 是 \TeX 组装水平盒子或竖直盒子时所处的模式。我们使用 *The \TeX book* 中的术语 “内部竖直模式” 来命名我们所认为的 “受限竖直模式”。请参阅 “” depth height (第 1 页) 和 “” depth height (第 1 页)。

¹⁹ \copy 'copy'

. rule) 是一个黑色实心矩形。标线和盒子一样也有宽度、高度和深度。该矩形的竖直尺寸是它的高度和深度之和。普通的水平或竖直线是标线的特殊情形。

标线可以是水平的或者竖直的。水平标线和竖直标线的区别与你如何生成标线有关，因为竖直标线可以矮而胖（从而看似水平标线），而水平标线也可以高而瘦（从而看似竖直标线）。 \TeX

```
\hrule \vrule \color{vrule} \hrule
```

```
\vrule height .25in' .25
```

- 1) \TeX \TeX
- 2)

```
\vrule height1pt depth2pt width3in'
```

The \TeX book 221–222 \TeX

. script size) 描述一族字体中三个相关字体的其中一个。标号尺寸比文本尺寸小但比小标号尺寸大。对上标，下标以及文本中分式的分子和分母， \TeX

. scriptscript size) 描述一族字体中三个相关字体的最小一个。对第二级的上标、下标、分子和分母， \TeX \TeX

. ‘

. \TeX space), 有如下几种方法:

- 1) 你可以写上某些 \TeX 视为空格记号的东西：一个或多个空格符，行尾符（行尾符作用和空格相似），或任何能展开为空格记号的命令。 \TeX 一般将多个连续空格视为单个空格，包括其中有一个行尾符的情形。（一个空行表示段落的结束；它导致 \TeX 生成一个 `\par` 记号。） \TeX 调整此种间隔的大小以适应由上下文所需的长度。
- 2) 你可以写上一个生成指定粘连的间距命令。粘连有伸长量或收缩量，从而生成或多或少的间隔。你可以有水平粘连和竖直粘连。在断行或分页位置之后的粘连将会消失。

- 3) 你可以写上一个紧排。紧排生成固定大小的间隔，它不可伸长也不可收缩，而且在断行或分页时也不会消失（除非其后紧跟着一个粘连）。紧排常用于给两个相邻盒子建立固定的空间关系。

粘连和紧排也可以取负值。两个相邻项目间的负粘连或负紧排将拉近它们俩的距离。

. “

. strut) 一个不可见的盒子，它的宽度为零，而高度和深度比上下文中的“正常”行稍微大一点。支架常用于在通常行间隔取消时获得一致的竖直间隔，比如在数学公式中，或者在设定了 `\offinterlineskip` 的水平阵列中。因为支架比该行的其他东西都高和深，它确定了该行的高度和深度。你可以用 `\strut` 命令（第 `\strut` 页）或 `\mathstrut` 命令（第 `\mathstrut` 页）生成一个支架。`\strut` 可以在任何地方使用，但 `\mathstrut` 只能在数学模式中使用。在 plain \TeX 中，支架的高度为 8.5 pt，深度为 3.5 pt

```
\vbox{\hsize = 3in \raggedright
  \strut Here is the first of two paragraphs that we're
  setting in a much narrower line length.\strut}
\vbox{\hsize = 3in \raggedright
  \strut Here is the second of two paragraphs that we're
  setting in a much narrower line length.\strut}
```

Here is the first of two paragraphs that
we're setting in a much narrower line length.

Here is the second of two paragraphs that
we're setting in a much narrower line length.

```
$\overline{x\mathstrut} \otimes \overline{t\mathstrut}$
```

\overline{x} \overline{t}

$\overline{x} \otimes \overline{t}$

· style) 的其中一种。对样式的了解,将有助于你将某部分公式设为不同类型的大小,即与 $\text{T}_{\text{E}}\text{X}$ 按照通用法则选择的大小不同。

其中的四种主要样式如下:

陈列样式	(用于单独一行显示的公式)
文本样式	(用于嵌入普通文本内的公式)
标号样式	(用于上标和下标)
小标号样式	(用于上标中的上标,等等)

另外四种样式被称为狭窄变体。在这些变体中,上标升高得没那么多,因而公式和原来相比占用更少的竖直间隔。在 *The $\text{T}_{\text{E}}\text{X}$ book* 第 140-141 页中详细介绍了 $\text{T}_{\text{E}}\text{X}$ 如何选择样式。

$\text{T}_{\text{E}}\text{X}$ 依照样式选择尺寸类型:

- 陈列样式和文本样式设定为文本尺寸,就像 ``this'` 这样。
- 标号样式设定为标号尺寸,就像 ``this'` 这样。
- 小标号样式设定为小标号尺寸,就像 ``this'` 这样。

参考 “” depth height (第 ``'` 页) 以获得更多关于这三种尺寸的信息。

在 $\text{T}_{\text{E}}\text{X}$ 中没有“小小标号”样式,因为此样式对应的尺寸太小以致无法阅读。因此 $\text{T}_{\text{E}}\text{X}$ 用小标号样式设定三阶上下标等。

偶尔你会发现 $\text{T}_{\text{E}}\text{X}$ 给公式设定的样式和你想要的不同。你可以用 `\textstyle`, `\displaystyle`, `\scriptstyle` 和 `\scriptscriptstyle` 命令 (第 ``textstyle'` 页) 改变 $\text{T}_{\text{E}}\text{X}$ 的选择。

$\text{T}_{\text{E}}\text{X}$ $\text{M}_{\text{E}}\text{X}$. (a) $\text{T}_{\text{E}}\text{X}$ (b) $\text{T}_{\text{E}}\text{X}$

· text math) 这个术语表示放在一行文本之中的数学公式,即两边用 $\$$ 括起来的公式。 $\text{T}_{\text{E}}\text{X}$ 在文内数学模式中排版文内公式。

· text size) 描述一族字体中三个相关字体的最大一个。对数学模式中的普通符号, $\text{T}_{\text{E}}\text{X}$

· token) 要么是一个标有类别码的字符,要么是一个控制序列。 $\text{T}_{\text{E}}\text{X}$ 用它的眼睛从文件中读取字符串 (见 “ $\text{T}_{\text{E}}\text{X}$ ” depth height, 第 ``texpouxi'` 页), 并用它的嘴将字符串咀嚼成一个个记号。当一个记号到达 $\text{T}_{\text{E}}\text{X}$ $\text{T}_{\text{E}}\text{X}$

· `\vbox`, vertical box) 是 $\text{T}_{\text{E}}\text{X}$ 从上到下逐个放置竖直列表的项目而构造出来的盒子。竖直盒子, 作为一个单元, 既不是本质上水平的也不是本质上竖直的, 即它可以出现在竖直列表或水平列表中。你可以用 `\vbox` 或 `\vtop` 命令 (第 `\vtop` 页) 构造竖直盒子。两者的区别在于, 对于 `\vbox`, 所构造的竖直盒子的基准点来源于列表的最后一个 (通常是最下面那个) 组成项; 而对于 `\vtop`, 所构造的竖直盒子的基准点来源于列表的第一个 (通常是最上面那个) 组成项。

· vertical list) 是 $\text{T}_{\text{E}}\text{X}$ 位于某种竖直模式时 (即组装竖直盒子或页面时) 生成的一系列项目。见下面的“竖直模式”。

· $\text{T}_{\text{E}}\text{X}$ vertical mode) 之一: 普通竖直模式用于组装主竖直列, 而内部竖直模式用于组装竖直盒子。只要 $\text{T}_{\text{E}}\text{X}$ 位于竖直模式中, 它的胃 (见“ $\text{T}_{\text{E}}\text{X}$ ” depth height, 第 `\texpouxi` 页) 就在构造项目 (盒子、粘连、惩罚等) 的竖直列表。 $\text{T}_{\text{E}}\text{X}$ 逐个排版列表中的项目, 从上到下。

竖直列表不能包含任何由本质上的水平命令, 比如 `\hskip` 或普通的 (非空格) 字符, 生成的项目。²⁰

- 如果 $\text{T}_{\text{E}}\text{X}$ 正在普通竖直模式中组装竖直列表, 并碰到一个本质上的水平命令, 它将切换到水平模式。
- 如果 $\text{T}_{\text{E}}\text{X}$ 正在内部竖直模式中组装竖直列表, 并碰到一个本质上的水平命令, 它将报错。

这两个命令你也许会认为是本质上竖直的, 实际上却是本质上水平的: `\valign` (第 `\valign` 页) 和 `\vrule` (第 `\vrule` 页)。在 *The $\text{T}_{\text{E}}\text{X}$ book* 第 283 页中列出了所有本质上的水平命令。

有个特别重要的事实需要知道, 即 $\text{T}_{\text{E}}\text{X}$ 将任何普通的非空格符视为本质上水平的。如果 $\text{T}_{\text{E}}\text{X}$ 在你预期之外突然开始一个新段落, 这可能是由 $\text{T}_{\text{E}}\text{X}$ 在竖直模式中遇到的字符引起的。要让 $\text{T}_{\text{E}}\text{X}$ 不把该字符视为本质上水平的, 你可以将它放入一个水平盒子中; 这是因为, 尽管看名字容易误解, `\hbox` 命令既不是本质上水平的也不是本质上竖直的。

· `\whatsit`) 是一个信息项目, 它用于让 $\text{T}_{\text{E}}\text{X}$ 执行某些无法归入普通事物体系中的操作。小玩意可以出现在水平列表或竖直列表中, 就像一个盒子或粘连项那样。 $\text{T}_{\text{E}}\text{X}$ 将小玩意排版为一个宽度、高度和深度都为零的盒子——即一个不包含任何东西也不占用任何空间的盒子。

在 $\text{T}_{\text{E}}\text{X}$ 中内建了三种类型的小玩意:

- `\openout`、`\closeout` 和 `\write` 命令 (第 `\openout` 页) 生成用于操作输出文件的小玩意。 $\text{T}_{\text{E}}\text{X}$ 将此操作推迟到下次送出页面到

²⁰ $\text{T}_{\text{E}}\text{X}$

.dvi 文件时（除非在该操作之前加上 `\immediate`）。TeX 用小玩意表示这些命令，因为它们与当时正在排版的内容完全无关。

- `\special` 命令（第 ``special'` 页）让 TeX 直接插入某些文本到 .dvi 文件中。如同 `\write` 命令，TeX 也将此操作推迟到下次送出页面到 .dvi 文件时。`\special` 的典型用法是命名一个图片文件，以让设备驱动文件将它整合到最终输出中。
- 当你用 `\language` 或 `\setlanguage` 命令（第 ``language'` 页）改变语言时，TeX 插入一个小玩意，以引导它在以后分段为行时使用特定集合的连字规则。

TeX 的某些特别实现可能提供额外的小玩意。

. width) 是它总共占用的水平间隔，即从它的左边缘到右边缘的距离。盒子内排版的素材可以比盒子本身还宽。





`'cmddesc'`



<code>\AA</code>	Å
<code>\aa</code>	å
<code>\AE</code>	Æ
<code>\ae</code>	æ
<code>\L</code>	Ł
<code>\l</code>	ł
<code>\O</code>	/挪威文字母
<code>\o</code>	/挪威文字母
<code>\OE</code>	Œ
<code>\oe</code>	œ
<code>\ss</code>	ß

¹ TeX (第 `'resources'` 页) TeX

T_EX

{\it les \oe vres de Moli\‘ere}

les œvres de Molière



☆ \# #²
\\$ \$
\% %
\& &
_ -
\lq ‘
\rq ’
\lbrack [
\rbrack]
\dag †
\ddag ‡
\copyright ©
\P ¶
\S §
T_EX (#, \$, %, &, _)

\dag \\$9.98, 但在英格兰要 {\it \\$}24.98。 =

† \$9.98, 但在英格兰要 £24.98。

\TeX
T_EX _

² ³ #。

A book about `\TeX` is in your hands.

A book about `TEX` is in your hands.

`\dots`

, `\ldots` `'\ldots'` `\dots\lrcorner`.

`x_1`, `x_2`, `\dots`, `x_{∞}`

$x_1, x_2, \dots, x_\infty$

“各种常用数学符号” (第 `'\specsyms'` 页).

■

`\char` $\langle charcode \rangle$

$\langle charcode \rangle$

`{\char65}` `{\char 'A}` `{\char '\A}`

A A A

`\mathchar` $\langle mathcode \rangle$

$\langle mathcode \rangle$

`\def\digger{\mathchar "027F} % plain TeX \spadesuit.`

`% 0 类, 2 族, "7F .`

`\digger`

♠

`\delimiter` `'\delimiter'`



`\'` `é`

`\.` `ñ`

`\=` `ṛ`

`\^` `ô`

`\‘` `è`

`\"` `ö`

`\~` `ã`

`\c` `ç`

`\d` `ṛ`

`\H` `ő`

`\t` `û`

`\u` `ř`

`\v` `ö`

“空格”, ‘`spaces`’

Add a soup\c con of \'elan to my pin\~a colada.

Add a soupçon of élan to my pinã colada.

`\i`

`\j`

‘i’ ‘j’. 在普通文本中, 如果需要为 ‘i’ ‘j’ ‘i’ ‘j’ `\imath` `\jmath` (第 `\imath` 页)。

long ‘i’ as in life `\i` fe `\quad` `\v` j

long ‘i’ as in life j

`\accent` `\langle charcode \rangle`

$\langle charcode \rangle$ \TeX 1ex。如果需要标记重音的字符过高或者过低, \TeX \TeX

```
l'H\accent94 otel des Invalides
% Position 94 of font cmr10 has a circumflex accent.
```

l'Hôtel des Invalides

`'mathaccent'`

■

$\backslash noboundary$

\TeX $\backslash noboundary$ \TeX \TeX \TeX $\backslash noboundary$ \TeX



`\fivebf` 5
`\fivei` 5
`\fiverm` 5
`\fivesy` 5
`\sevenbf` 7
`\seveni` 7
`\sevenrm` 7
`\sevensy` 7
`\tenbf` 10
`\tenex` 10
`\teni` 10
`\tenrm` 10
`\tensl` 10
`\tensy` 10
`\tenit` 10
`\tentt` 10

TeX

See how I've reduced my weight---from
 120 lbs.\ to {\sevenrm 140 lbs}.

See how I've reduced my weight—from 120 lbs. to 140 lbs.

`\nullfont`

TeX TeX



☆ \bf
 \it
 \rm
 \sl
 \tt

3

“数学公式字体”（第 `\mathfonts` 页）。

\lccode $\langle charcode \rangle$ [$\langle number \rangle$ table entry]
 \uccode $\langle charcode \rangle$ [$\langle number \rangle$ table entry]

\lccode 和 \uccode 的值为256个输入字符设定大小写的对应关系。
 \lowercase 和 \uppercase 命令以及 T_EX 的断词算法将会使用这些值。

T_EX 这样初始化 \lccode和\uccode的值：

- 小写字母的\lccode是它的ASCII代码。
- 大写字母的\lccode是它的小写形式的ASCII代码。
- 大写字母的\uccode是它的ASCII代码。
- 小写字母的\uccode是它的大写形式的ASCII代码。
- 非字母字符的\lccode和\uccode都是零。

³ typeface) 和字体 (font), 字样指一整套统一设计的字体。 ⁴

The Dormouse was {\it not} amused.

The Dormouse was *not* amused.

在大多数时候，没有必要改变这些值，除非你所使用语言的字母比英文多。

```
\char\uccode's \char\lccode'a \char\lccode'M
```

Sam

```
\lowercase { \token list }
```

```
\uppercase { \token list }
```

这些命令按照`\lccode`和`\uccode`的值转换 $\langle token\ list \rangle$ 中的字母（类别码为11的）为它们的大写和小写形式。非字母字符不受影响，即使它们是宏调用或其他展开成字母的命令。

```
\def\x{Cd} \lowercase{Ab\x} \uppercase{Ab\x}
```

abCd ABCd

☆ _

这个命令明确地产生一个单词间空格，称为“控制空格”。当一个字母出现在一个控制序列之后或者其他情况下你不想让两个记号排在一起时，就要用到控制空格。由`_`产生的空白不受前面标点的影响，即其距离因子（第`'spacefactor'`页）为1000。

如果你想打印出`'_'`空格，可以输入`{\tt \char '\ }`。

```
The Dormouse was a \TeX\ expert, but he never let on.
```

The Dormouse was a T_EX expert, but he never let on.

```
\space
```

这个命令等价于一个输入的空格字符。和`\`不同，它的宽度受到前面标点的影响。

```
Yes.\space No.\space Maybe.\par
Yes.\_No.\_Maybe.
```

```
Yes. No. Maybe.
Yes. No. Maybe.
```

^^M

这将产生行尾字符。通常有两个效果：

- 1) 类似一个命令，产生一个输入的空格（如果它出现在一个非空行的尾部）或者一个 `\par` 记号（如果出现在空行的尾部）。
- 2) 结束一行，使 `TeX` 忽略剩余的字符。

但是，如果以 `'^^M`（即 control-M 的 ASCII 代码 13）出现，`^^M` 结束一行。你可以修改类别码，从而赋予它新的含义。关于 ^^ 更多的解释请参见第 `'twocarets'` 页。

```
Hello.^^MGoodbye.
Goodbye again.\par
The \char '\^^M\ character.\par
% The fl ligature is at position 13 of font cmr10
\number '\^^M\ is the end of line code.\par
Again, \number '\^^M is the end of line code,
isn't it? % 32 is the ASCII code for a space
```

```
Hello. Goodbye again.
The fl character.
13 is the end of line code.
Again, 32isn't it?
```

☆ ~

活动字符 `'~'` 称为 “带子” (tie)，它产生一个正常的单词间空白，同时保证前后两个单词之间不会断行。只要断行会有歧义，就应该使

用带子。比如，在中间名之前，缩写词 ``Dr." 之后，或者 ``Fig." 之后：像 ``Fig. 8"。

```
P.D.Q.~Bach (1807--1742), the youngest and most
imitative son of Johann~S. Bach, composed the
{\sl Concerto for Horn and Hardart}.
```

P.D.Q. Bach (1807–1742), the youngest and most imitative son of
Johann S. Bach, composed the *Concerto for Horn and Hardart*.

☆ \/

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 字体中每个字符都有 ``倾斜修正"。直立字体的倾斜修正一般就是零。当从倾斜的字体（不一定是意大利体）切换到直立的字体时，倾斜修正会增加额外的空白。这个额外的空白之所以必要，是因为倾斜的字符略微挤占了后面的空格位置，当后面跟着直立字体时，会显得中间的空格太小。所有字符的倾斜修正都包含在字体的度量文件里。

命令`\/`产生对前面字符的倾斜修正。当从倾斜字体切换到直立字体时，必须要插入倾斜修正，除非下一个字符是句点或逗号。

```
However, {\it somebody} ate {\it something}: that's clear.
```

```
However, {\it somebody\/} ate {\it something\/}:
that's clear.
```

However, *somebody* ate *something*: that's clear.

However, *somebody* ate *something*: that's clear.

```
\frenchspacing
\nonfrenchspacing
```

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 一般会为了标点符号而调整单词间距。例如，一个句子的末尾会有额外的空白，句尾标点符号之后的粘连会被拉长。`\frenchspacing`命令让 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 调整单词间距时忽略标点符号；`\nonfrenchspacing`命令让 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 使用正常间距，即不使用`\frenchspacing`效果。

关于 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 如何处理句尾标点，请参考第 `\periodspacing` 页。

```
{\frenchspacing An example: two sentences. Right? No.\par}
{An example: two sentences. Right? No. \par}%
```

An example: two sentences. Right? No.

An example: two sentences. Right? No.

`\obeyspaces`

`\TeX` 把连续的空格处理为一个空格。`\obeyspaces`的作用是输入多少空格，就输出多少。但是`\obeyspaces`不会显示行首的空格。这时候推荐使用`\obeywhitespace`，其定义在`eplain.tex`中(第`'ewhitesp'`页)。`\obeyspaces`主要用于以等宽字体打印计算机代码，以及显示所见即所得的内容。

`\obeylines`命令(第`'obeylines'`页)的作用是让`\TeX`输出跟你所输入的一模一样的行。`\obeylines`经常和`\obeyspaces`一起被使用。

```
These      spaces      are      closed      up
{\obeyspaces but      these are      not      }.
```

These spaces are closed up but these are not .

```
\spacefactor    [ <number> parameter ]
\spaceskip      [ <glue> parameter ]
\xspaceskip     [ <glue> parameter ]
\sffcode <charcode>    [ <number> table entry ]
```

这些基本参数影响相邻单词之间的空白，即单词间距。默认的单词间距由当前字体决定。当`\TeX`处理一个水平列表时，会监视`\spacefactor`间隔因子 f 。每当一个输入字符 c 被处理时， f 就会随 f_c (c 的间隔因子代码)的值而更新。大多数的字符， f_c 是1000，因此`\TeX`设定 f 为1000。(f 的初始值也是1000. `depth height`) 当`\TeX`遇到一个单词间空格的时候，就会调整间隔的大小，给该间隔的伸长量和收缩量分别乘以 $f/1000$ 和 $1000/f$ 。所以：

- 1) 当 $f = 1000$ 时，单词间距为默认值。
- 2) 当 $f < 1000$ 时，单词间距有更少伸长量更多收缩量。
- 3) 当 $f > 1000$ 时，单词间距有更多伸长量更少收缩量。

另外，如果 $f \geq 2000$ ，当前字体的“额外空白”参数会让单词间距进一步增大。

每一个输入字符 c 对应于`\sfcode`（间隔因子代码）表中的一项。`\sfcode`表项与字体无关。通常 $\text{T}_{\text{E}}\text{X}$ 处理完 c 之后就令 f 等于 f_c 。但是：

- 如果 f_c 为零， f 将不变。所以 plain $\text{T}_{\text{E}}\text{X}$ 中 f_c 为零的字符，如 `\'`，对于单词间距的计算毫无影响。
- 如果 $f < 1000 < f_c$ ， $\text{T}_{\text{E}}\text{X}$ 会让 f 为1000而不是 f_c ，也就是，不会让 f 快速地增加。

句号的`\sfcode`值通常是3000，所以 $\text{T}_{\text{E}}\text{X}$ 会在句号之后增加额外的空白。（参见上面 $f \geq 2000$ 的情形）。水平列表中的非字符，比如竖直标线，一般视为间隔因子为 1000 字符。

通过修改 `\spacefactor` 的值，就可以显式地调整间隔因子。通过修改 `\xspaceskip` 或者 `\spaceskip` 的值，你还可以覆盖默认的单词间距：

- `\xspaceskip` 设定 $f \geq 2000$ 时的粘连；如果 `\xspaceskip` 是零，就取默认值。
- `\spaceskip` 设定 $f < 2000$ 或者 `\xspaceskip` 为零时的粘连；如果 `\spaceskip` 是零，就取默认值。`\spaceskip` 粘连的伸长或收缩依 f 的值而变，就像普通的单词间距一样。

The $\text{T}_{\text{E}}\text{X}$ book 第 76 页 对 $\text{T}_{\text{E}}\text{X}$ 计算单词间粘连的规则有一个具体的解释。*The $\text{T}_{\text{E}}\text{X}$ book* 第 285--287 页详细描述了水平列表的各种项目之后`\spacefactor`的调整。

☆ `\centerline` $\langle argument \rangle$

`\leftline` $\langle argument \rangle$

`\rightline` $\langle argument \rangle$

命令 `\centerline` 产生一个与当前行同宽的水平盒子，并把 $\langle argument \rangle$ 在里面居中放置。命令 `\leftline` 和 `\rightline` 的作用类似：把 $\langle argument \rangle$ 放在盒子里靠左或者靠右的位置。如果想让几行内容都有同样的效果，每一行都必须这样操作。其他的办法，请看第 `\eplaincenter` 页。

这些命令不能用在段落里，否则分段成行的时候， $\text{T}_{\text{E}}\text{X}$ 可能会遇到麻烦并警告盒子溢出了。

```
\centerline{Grand Central Station}
\leftline{left of Karl Marx}
\rightline{right of Genghis Khan}
```

Grand Central Station

left of Karl Marx

right of Genghis Khan

☆ \line ⟨*argument*⟩

这个命令产生一个 hbox 来包含⟨*argument*⟩。hbox 的宽度和当前行宽相同，即从左边距延伸到右边距。

```
\line{ugly \hfil suburban \hfil sprawl}
% Without \hfil you'd get an 'underfull box' from this.
```

ugly suburban sprawl

\llap ⟨*argument*⟩

\rlap ⟨*argument*⟩

这些命令把文本覆盖到当前位置的左边或者右边。\\llap会向左退⟨*argument*⟩的宽度，然后开始排版⟨*argument*⟩。\\rlap类似，但是先排版⟨*argument*⟩然后把后面的内容向左退。\\llap和\\rlap主要用于把文本放在现在的页边距之外。它们的工作原理是创建一个零宽度的盒子。

\\llap和\\rlap可以被用来以叠印的方式新建特殊字符。但是必须确保作原料的字符一般宽（等宽字体如cmtt10和 Computer Modern 10-point 的打字机字体是这样的）。

```
\noindent\llap{off left }\line{\vrule $\Leftarrow$
left margin of examples\hfil right margin of examples
$\Rightarrow$\vrule}\rlap{ off right}
```

off left |← left margin of examples

right margin of examples ⇒| off right

\hsize (第 `\hsize` 页).

■ `\par`

这个命令结束一个段落，并使 `TEX` 进入竖直模式来在页面上增加更多内容。因为 `TEX` 把一个空行当作 `\par` 记号，所以不需要明确地键入 `\par`。

必须指出 `\par` 不会让 `TEX` 开始新的一段；它仅仅告诉 `TEX` 去结束一段。要使 `TEX` 开始新段落，需要处于竖直模式之下（在 `\par` 之后已经是了），并遇到一个水平项目比如一个字母。为了开始新段落，`TEX` 将插入一些竖直空白，取决于 `\parskip` (第 `\parskip` 页)；新段落缩进，取决于 `\parindent` (第 `\parindent` 页)。

要想减小段落间 `\par` 产生的空白，通常可以使用命令 `\vskip -\lastskip`。■
当你需要写一个宏而让它与前面是不是空行无关的时候，这个命令就有用。

你可以使用 `\everypar` (第 `\everypar` 页) 来让 `TEX` 在每开始一个新段落的时候做一些事。

`\par` 的具体效果请参阅 *The T_EXbook* pages 283 and 286。

```
\parindent = 2em
“Can you row?” the Sheep asked, handing Alice a pair of
knitting-needles as she was speaking.\par “Yes, a little%
---but not on land---and not with needles---” Alice was
starting to say, when suddenly the needles turned into oars.■
```

“Can you row?” the Sheep asked, handing Alice a pair of
knitting-needles as she was speaking.

“Yes, a little—but not on land—and not with needles—” Alice
was starting to say, when suddenly the needles turned into oars.

`\endgraf`

这个命令与 `\par` 同义。有时候你重新定义了 `\par`，但是又想取得 `\par` 的原始定义，这个命令就会有用。

`\parfillskip` [*glue* parameter]

这个参数设定 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 插入段落末尾的水平粘连。默认值是 `0pt plus 1fil`，这使得最后一行被空格填满。如果设成 `0pt`， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 就会把段落的最后一行拉伸到右边距。

☆ `\indent`

当 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 处于竖直模式时，比如段落结束以后，这个命令插入段落间粘连 `\parskip`，使 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 进入水平模式，开始新的一段，并缩进 `\parindent` 大小。如果 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 已经处于水平模式了，这个命令就只做缩进。一行中两个 `\indent` 产生两个缩进。

如下面的例子所示，新的段落开始之后的 `\indent` 是多余的。当 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 处在竖直模式下，遇到一个字母或其他水平模式中的命令时，就会切换到水平模式缩进 `\indent`，然后继续。

```
\parindent = 2em This is the first in a series of three
paragraphs that show how you can control indentation. Note
that it has the same indentation as the next paragraph.\par
\indent This is the second in a series of three paragraphs.
It has \indent an embedded indentation.\par
\indent\indent This doubly indented paragraph
is the third in the series.
```

This is the first in a series of three paragraphs that show how you can control indentation. Note that it has the same indentation as the next paragraph.

This is the second in a series of three paragraphs. It has an embedded indentation.

This doubly indented paragraph is the third in the series.

☆ `\noindent`

当 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 结束一段后处于竖直模式时，这个命令插入段落间粘连 `\parskip`，使 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 进入水平模式，然后开始新段，不要缩进。在水平模式中，比如段落中，这个命令没有效果。`\noindent` 开始的新段会取消通常的 `\parindent` 缩进。

当新的段落是在某些独立显示的内容之后出现时，`\noindent`通常用于取消段落第一行的缩进。

```
\parindent = 1em
Tied round the neck of the bottle was a label with the
words \smallskip \centerline{EAT ME}\smallskip
\noindent beautifully printed on it in large letters.
```

Tied round the neck of the bottle was a label with the words
EAT ME
beautifully printed on it in large letters.

`\textindent` *<argument>*

这个命令让 TeX 开始新段，正常缩进 `\parindent`。然后 TeX 在缩进的地方向右对齐排版 *<argument>* 并加入一个 en 大小的空白（em 的一半）。Plain TeX 使用这个命令来排版脚注（第 `'footnote'` 页）以及列表项（见 `\item`，depth height plus 0.1em minus 0.1 em 第 `'item'` 页）。

```
\parindent = 20pt \textindent{\raise 1pt\hbox{$\bullet$}}%
You are allowed to use bullets in \TeX\ even if
you don't join the militia, and many peace-loving
typographers do so.
```

- You are allowed to use bullets in TeX even if you don't join the militia, and many peace-loving typographers do so.

`\parindent` [*<dimen>* parameter]

这个参数设定段落第一行缩进的大小。如下面的例子所示，最好不要把 `\parindent` 和 `\parskip` 都设为零，否则两段的区别就不明显了。

```
\parindent = 2em This paragraph is indented by 2 ems.
\par \parindent=0pt This paragraph is not indented at all.
\par Since we haven't reset the paragraph indentation,
this paragraph isn't indented either.
```

This paragraph is indented by 2 ems.
 This paragraph is not indented at all.
 Since we haven't reset the paragraph indentation, this paragraph
 isn't indented either.

`\everypar` [*<token list>* parameter]

TeX 处于水平模式下，比如新段开始的时候，就会在`()`中执行这个命令。默认的`\everypar`为空，但是你可以添加一些命令，每一段开始的时候在记号列表中执行 并为`\everypar`分配记号列表。

```
\everypar = {$\Longrightarrow$\enspace}
Now pay attention!\par
I said, ‘‘Pay attention!’’.\par
I'll say it again! Pay attention!
```

⇒ Now pay attention!
 ⇒ I said, “Pay attention!”.
 ⇒ I'll say it again! Pay attention!



☆ `\hsize` [*<dimen>* parameter]

这个参数设定当前的行宽，即从左边距开始算起，段落里一行的宽度。很多 TeX 命令，比如`\centerline` (第 `'centerline'` 页) 和`\hrule` (第 `'hrule'` 页)，都间接使用了`\hsize`的值。改变`\hsize`的值，就相应地改变了这些命令的宽度效果。

如果`\hsize`的变化发生在一个非空的竖直盒子里，盒子的宽度就是给定的`\hsize`。

Plain T_EX 的默认\hsize为6.5in。

```
{\hsize = 3.5in % Set this paragraph 3.5 inches wide.
The hedgehog was engaged in a fight with another hedgehog,
which seemed to Alice an excellent opportunity for
croqueting one of them with the other.\par}%
```

The hedgehog was engaged in a fight with another hedgehog, which seemed to Alice an excellent opportunity for croqueting one of them with the other.



```
\leftline{\raggedright\top{\hsize = 1.5in
Here is some text that we put into a paragraph that is
an inch and a half wide.}\quad
\top{\hsize = 1.5in Here is some more text that
we put into another paragraph that is an inch and a
half wide.}}
```

Here is some text that we put into a paragraph that is an inch and a half wide.	Here is some more text that we put into another paragraph that is an inch and a half wide.
--	---

☆ \narrower

这个命令产生较窄的段落，左右边距将会增加 \parindent（即当前段落缩进的大小），原因是\leftskip和\rightskip分别增加了\parindent。■
通常\narrower被放在一个编组的开始，作用于其中的段落上。如果你忘记把\narrower封闭在编组里，就会发现文档所余部分全都变窄了。

\narrower只影响其后的段落。如果包含\narrower的编组在段落末尾之前结束，T_EX 将不会产生窄的段落。

```
{\parindent = 12pt \narrower\narrower\narrower
This is a short paragraph. Its margins are indented
three times as much as they would be
had we used just one ‘‘narrower’’ command.\par}
```

This is a short paragraph. Its margins are indented three times as much as they would be had we used just one “narrower” command.

```
\leftskip [⟨glue⟩ parameter]
\rightskip [⟨glue⟩ parameter]
```

这些参数告诉 T_EX 在当前段落每一行的左右各放多大的空白。我们将只解释 `\leftskip`，因为 `\rightskip` 与其类似。

你可以通过设置 `\leftskip` 为一个固定的非零尺寸来增加左边距。如果 `\leftskip` 是一个粘连，就会得到右对齐的内容，即左边距参差不齐。

通常，你应该在一个编组里为 `\leftskip` 赋值，来避免影响其余的文档部分。可是在一个段落内的编组里改变 `\leftskip` 的值是无意义的——只有段落处 `\leftskip` 的值才会被 T_EX 用来拆段成行。

```
{\leftskip = 1in The White Rabbit trotted slowly back
again, looking anxiously about as it went, as if it had
lost something. {\leftskip = 10in % has no effect
It muttered to itself, ‘‘The Duchess! The Duchess! She’ll
get me executed as sure as ferrets are ferrets!’’}\par}%

```

The White Rabbit trotted slowly back again, looking anxiously about as it went, as if it had lost something. It muttered to itself, “The Duchess! The Duchess! She’ll get me executed as sure as ferrets are ferrets!”

```

\pretolerance = 10000 % Don't hyphenate.
\rightskip = .5in plus 2em
The White Rabbit trotted slowly back again, looking
anxiously about as it went, as if it had lost something.
It muttered to itself, ‘‘The Duchess! The Duchess! She’ll
get me executed as sure as ferrets are ferrets!’’

```

The White Rabbit trotted slowly back again, looking anxiously about as it went, as if it had lost something. It muttered to itself, “The Duchess! The Duchess! She’ll get me executed as sure as ferrets are ferrets!”

☆ `\raggedright`
`\ttraggedright`

这些命令让 $\text{T}_{\text{E}}\text{X}$ 以“左对齐” `depth height plus 0.1em minus 0.1em` 的方式排版文档，其中单词间距保持正常，即间距一样，不需伸缩。因此右边距是不均匀的。而 $\text{T}_{\text{E}}\text{X}$ 默认的则是均匀对齐，即左右边距一样。均匀对齐的文本单词间距将会有伸缩以产生均匀的右边距。有些排版者偏好使用左对齐。因为左对齐避免了页面上的空白参差不齐。

当使用等宽字体排版的时候，你需要用 `\ttraggedright` 命令；而使用其他字体时，需要用 `\raggedright` 命令。

通常你会希望对整篇文档使用这些命令，但是通过编组就可以把效果局限在部分文本中。

```
\raggedright ‘‘You couldn’t have it if you {\it did\}
want it,’’ the Queen said. ‘‘The rule is, jam tomorrow
and jam yesterday---but never jam {\it today\}.’’
‘‘It {\it must\} come sometimes to ‘jam today,%
thinspace’’ Alice objected. ‘‘No, it can’t’’, said the
Queen. ‘‘It’s jam every {\it other\} day: today isn’t
any {\it other\} day.’’
```

“You couldn’t have it if you *did* want it,” the Queen said. “The rule is, jam tomorrow and jam yesterday—but never jam *today*.” “It *must* come sometimes to ‘jam today,’” Alice objected. “No, it can’t”, said the Queen. “It’s jam every *other* day: today isn’t any *other* day.”

`\hang`

此命令使段落第二行开始各行也缩进 `\parindent` 大小 (第 `\parindent'` 页)。■
因为第一行已经缩进了 `\parindent` (除非你使用 `\noindent` 取消了缩进), 所以整个段落呈现悬挂缩进。

```
\parindent=24pt \hang ‘‘I said you {\it looked} like an
egg, Sir,’’ Alice gently explained to Humpty Dumpty. ‘‘And
some eggs are very pretty, you know,’’ she added.
```

“I said you *looked* like an egg, Sir,” Alice gently explained to Humpty Dumpty. “And some eggs are very pretty, you know,” she added.

`\hangafter` [`<number>` parameter]

`\hangindent` [`<dimen>` parameter]

这两个参数一起决定段落的“悬挂缩进” depth height。悬挂缩进使 T_EX 产生某些行缩进而其他行依然正常的效果。`\hangafter` 决定哪些行将被缩进; `\hangindent` 决定缩进的大小以及位置是左边还是右边:

- 设 `\hangafter` 的值为 n 。如果 $n < 0$, 开始的 $-n$ 行将被缩进。如果 $n \geq 0$, 除开始的 n 行以外的各行将被缩进。
- 设 `\hangindent` 的值为 x 。如果 $x \geq 0$, 左边将缩进 x ; 如果 $x < 0$, 右边将缩进 $-x$ 。

当你设置悬挂缩进的时候，仅对下一段（如果处于竖直模式）或者当前段（如果处于水平模式）有效。TeX拆段成行时，使用的是段落末尾的`\hangafter`和`\hangindent`的值。

与其他塑段参数不同的是，`\hangafter`和`\hangindent`在每一段的开始都被重设为其默认值，即`\hangafter`为1，`\hangindent`为0。想要排版许多悬挂缩进的段落，可以使用`\everypar`（第‘`everypar`’页）。如果你同时设定`\hangafter`，`\hangindent`以及`\parshape`，TeX将忽略`\hangafter`和`\hangindent`。

```
\hangindent=6pc \hangafter=-2
```

This is an example of a paragraph with hanging indentation.■
In this case, the first two lines are indented on the left,■
but after that we return to unindented text.

This is an example of a paragraph with hanging in-
dentation. In this case, the first two lines are in-
dented on the left, but after that we return to unindented text.

```
\hangindent=-6pc \hangafter=1
```

This is another example of a paragraph with hanging indentation. Here, all lines after the first have been indented on the right. The first line, on the other hand, has been left unindented.

This is another example of a paragraph with hanging indentation. Here, all lines after the first have been indented on the right. The first line, on the other hand, has been left unindented.

```
\parshape n i_1 l_1 i_2 l_2 ... i_n l_n
```

此命令指定一个段落前 n 行的形状。如果处于水平模式，该命令作用于当前段落；如果处于竖直模式，则作用于下一段落。其中的各个 i 和 l 都是尺寸。第1行的缩进为 i_1 ，长度为 l_1 ；第2行的缩进为 i_2 ，长度为 l_2 ；依此类推。若该段落超过 n 行，后面各行都使用 i_n 和 l_n 。为达到这里展示的特殊效果，你通常要反复调试，插入多个紧排，并且选择适合该形状的词语。

与 `\hangafter` 和 `\hangindent` 一样, `\parshape` 也仅对此段落有效。如果你同时指定 `\hangafter` 和 `\hangindent` 以及 `\parshape`, `TeX` 将忽略 `\hangafter` 和 `\hangindent`。


```
% A small font and close interline spacing make this work
\smallskip\font\sixrm=cmr6 \sixrm \baselineskip=7pt
\fontdimen3\font = 1.8pt \fontdimen4\font = 0.9pt
\noindent \hfuzz 0.1pt
\parshape 30 0pt 120pt 1pt 118pt 2pt 116pt 4pt 112pt 6pt
108pt 9pt 102pt 12pt 96pt 15pt 90pt 19pt 84pt 23pt 77pt
27pt 68pt 30.5pt 60pt 35pt 52pt 39pt 45pt 43pt 36pt 48pt
27pt 51.5pt 21pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt■
16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt■
53pt 14.6pt 48pt 24pt 45pt 30.67pt 36.5pt 51pt 23pt 76.3pt
The wines of France and California may be the best
known, but they are not the only fine wines. Spanish
wines are often underestimated, and quite old ones may
be available at reasonable prices. For Spanish wines
the vintage is not so critical, but the climate of the
Bordeaux region varies greatly from year to year. Some
vintages are not as good as others,
so these years ought to be
s\kern -.1pt p\kern -.1pt e\kern -.1pt c\hfil ially
n\kern .1pt o\kern .1pt t\kern .1pt e\kern .1pt d\hfil:
1962, 1964, 1966. 1958, 1959, 1960, 1961, 1964,
1966 are also good California vintages.
Good luck finding them!
```

```
The wines of France and California
may be the best known, but they
are not the only fine wines. Span-
ish wines are often underestimated,
and quite old ones may be avail-
able at reasonable prices. For
Spanish wines the vintage is
not so critical, but the cli-
mate of the Bordeaux re-
gion varies greatly from
year to year. Some
vintages are not as
good as others,
so these years
ought to be
specially
noted:
1962,
1964,
1966.
1958,
1959,
1960,
1961,
1964,
1966
are also
good Cal-
ifornia vintages.
Good luck finding them!
```

`\prevgraf` [*<number>* parameter]

在水平模式中，此参数表示该段落的当前行数；在竖直模式中，它表示上个段落的行数。只有在完成一些文本断行后，例如在陈列公式前或者在段落结尾处，`TeX` 才会设置 `\prevgraf`。详见 *The TeXbook* 第 103 页。

`\vadjust {<vertical mode material>}`

此命令在当前位置所在行下边插入指定的 *<vertical mode material>*。例如，你可以用它输出页面，或者在特定行添加额外空白。

```
Some of these words are \vadjust{\kern8pt\hrule} to be
found above the line and others are to be found below it.
```

```
Some of these words are to be found above the line and others are


---


to be found below it.
```

`\parindent` (第 `'parindent'` 页), `\parskip` (第 `'parskip'` 页), `\everypar` (第 `'everypar'` 页).

■

`\break`

强制在此处断行。除非用某种方法填满该行，否则你将得到 `“underfull hbox”` 的警告。此命令也能用于竖直模式。

```
Fill out this line\hfil\break and start another one.\par
% Use \hfil here to fill out the line.
This line is underfull---we ended it\break prematurely.
% This line causes an ‘underfull hbox’ complaint.
```

Fill out this line
and start another one.
This line is underfull—we ended it
prematurely.

`\nobreak`

阻止在此处断行。此命令也能用于竖直模式。

Sometimes you'll encounter a situation where
a certain space\nobreak\quad must not get lost.

Sometimes you'll encounter a situation where a certain space must
not get lost.

`\allowbreak`

在通常不断行的地方允许 \TeX 断行。它经常用于数学公式中，因为 \TeX 通常不愿意在其中断行。此命令也能用于竖直模式。

Under most circumstances we can state with some confidence
that $2+2\allowbreak=4$, but skeptics may disagree.
\par For such moronic automata, it is not difficult to
analyze the input/\allowbreak output behavior in the limit.■

Under most circumstances we can state with some confidence that
 $2 + 2 = 4$, but skeptics may disagree.
For such moronic automata, it is not difficult to analyze the input/
output behavior in the limit.

`\penalty <number>`

此命令生成一个惩罚 (penalty) 项。惩罚项使得 \TeX 或多或少愿意在此处断行。负惩罚值，实际上是奖励值，鼓励断行；正惩罚值阻碍断

行。大于或等于 10000 的惩罚值彻底阻止断行，而小于或等于 -10000 的惩罚值强制断行。此命令也能用于竖直模式。

```
\def\break{\penalty -10000 } % as in plain TeX
\def\nobreak{\penalty 10000 } % as in plain TeX
\def\allowbreak{\penalty 0 } % as in plain TeX
```

\obeylines

TeX 通常将行结束符视为空格。 \obeylines 让 TeX 将每个行结束符都视为段落结束符，从而强制断行。在排版诗歌或者程序代码时 \obeylines 大有用处。然而，如果某些行的长度超过行的实际长度 (`\hsize-\parindent`) depth height，它们将会被自动断行。

TeX 将在设定了 \obeylines 的各行间插入 \parskip (第 [\parskip](#) 页) 粘连 (既然它认为每行都是一个段落)。因此，在使用 \obeylines 时一般应该设定 \parskip 为零。

用 \obeyspaces 命令 (第 [\obeyspaces](#) 页) 可以让 TeX 按原样处理行中的空格。我们经常一起使用 \obeylines 和 \obeyspaces。

```
\obeylines
‘‘Beware the Jabberwock, my son!
\quad The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
\quad The frumious Bandersnatch!’’
```

```
“Beware the Jabberwock, my son!
  The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
  The frumious Bandersnatch!”
```

☆ \slash

此命令生成一个斜线符号 (/)，并且告诉 TeX 必要时可以在该斜线后断行。

```
Her oldest cat, while apparently friendly to most people,
had a Jekyll\slash Hyde personality when it came to mice.
```

Her oldest cat, while apparently friendly to most people, had a Jekyll/Hyde personality when it came to mice.

■

`\pretolerance` [*<number>* parameter]

`\tolerance` [*<number>* parameter]

在 $\text{T}_{\text{E}}\text{X}$ 选择段落断行点时，这两个参数确定对各行劣度 (badness) 的容许值 (tolerance)。劣度衡量了单词间距和理想情形差别多大⁵。其中 `\pretolerance` 指定在不加连字符断行时对劣度的容许值，而 `\tolerance` 指定加上连字符断行时对劣度的容许值。当某行排得太紧密 (单词间距过小)，或者排得太松散 (单词间距过大) 时，劣度就会超过容许值。

- 如果 $\text{T}_{\text{E}}\text{X}$ 将某一行排得太松散，它将给出 ```underfull hbox''` 的警告。
- 如果 $\text{T}_{\text{E}}\text{X}$ 将某一行排得太紧密，它将让该行超出右边缘，并给出 ```overfull hbox''` 的警告。

$\text{T}_{\text{E}}\text{X}$ 按照如下步骤选择断行点：

- 1) 它试图选择无需添加连字符的一系列断点。如果结果中每行的劣度都不超过 `\pretolerance`，这组断点是可接受的，该段落也就完成了断行。
- 2) 否则，它在允许添加连字符的情况下再次寻找断点。如果结果中每行的劣度都不超过 `\tolerance`，这组新断点是可接受的，该段落也就完成了断行。
- 3) 否则，它给每行的伸长量增加 `\emergencystretch` (见下面)，然后重新尝试。
- 4) 如果上述各种尝试都无法得到可接受的断点系列，它给该段落设置一个或多个溢出的水平盒子，并给出警告。

Plain $\text{T}_{\text{E}}\text{X}$ 设置 `\tolerance` 为 200，`\pretolerance` 为 100。若你设置 `\tolerance` 为 10000， $\text{T}_{\text{E}}\text{X}$ 将有无穷大的容许值，可以接受任何空隙，不管它有多糟糕。(除非它遇到即使添加连字符也无法放在一行的单词)。因此，通过改变 `\tolerance` 值，你可以避免溢出的或者松散的水

平盒子，但会得到更糟糕的空隙。通过增大 `\pretolerance` 值你可以让 `TEX` 不添加连字符（并且运行得更快），但同样可能得到更糟糕的空隙。而如果你设置 `\pretolerance` 为 `-1`，`TEX` 将不再尝试不添加连字符的断行方式。

参数 `\hbadness`（第 `'\hbadness'` 页）确定劣度达到多大时 `TEX` 才给出警告，但它不影响 `TEX` 对文档的排版。参数 `\hfuzz`（第 `'\hfuzz'` 页）确定水平盒子宽度超出多大时，`TEX` 才认为是错误的。

`\emergencystretch` [*<dimen>* parameter]

设定此参数大于零，可以让 `TEX` 更容易排版文档，而不是生成溢出的水平盒子。这样比设定 `\tolerance=10000` 更好，因为那样往往得到十分难看的行。如果 `TEX` 排版段落时无法不超过 `\tolerance` 值，它将给每行的伸长量增加 `\emergencystretch` 然后重新尝试。伸长量的增加将会缩减各行劣度，允许 `TEX` 生成比原来更宽的间距，从而选出当前情况下尽可能好的断行点。

`\looseness` [*<number>* parameter]

此参数用于修改段落的行数，相对其最佳行数。之所以称为松散度 (`\looseness`) depth height，是因为它衡量该段落有多松散，即包含多少额外的间距。

一般地，若 `\looseness` 等于 0，`TEX` 将按通常方式选择断行点。但若 `\looseness` 等于，比如 3，`TEX` 依下面步骤处理：

- 1) 按通常方式选择断行点，得到总行数为 n 的段落。
- 2) 丢弃这些断行点，并试着寻找总行数为 $n + 3$ 的一系列新断点。（缺少上一步，`TEX` 就无法知道 n 的值。）
- 3) 如果上一步尝试无法得到行劣度不超过 `\tolerance` 的结果，它将试着将段落分为 $n + 2$ 行；如果这也不行，试着分为 $n + 1$ 行；最终只能再次分为 n 行。

类似地，若松散度为 $-k$ ，`TEX` 试着让段落的行数比正常情形少 k 行。让段落多出一行的最简单方法是，将一个单词放到多出的那行。在最后两个单词之间加上一个带子 (tie, 第 `'@not'` 页) 符号，你就可以阻止此种断行方法。

设定 `\looseness` 是迫使段落占用给定行数的最好方法。想增加页面中的文本时，你可以将它设为负值。同样地，想减少页面中的文本时，可以将它设为正值。

在结束段落并分段为行之后， $\text{T}_{\text{E}}\text{X}$ 将 `\looseness` 重置为 0。若要改动多个段落的松散度，你必须对每个段落分别设置，或者将改动放在 `\everypar`（第 ``everypar'` 页）命令中。

`\linepenalty` [*$\langle number \rangle$* parameter]

此参数设定 $\text{T}_{\text{E}}\text{X}$ 分段为行时各行的惩罚值。此惩罚值与断行位置无关。增大此参数将让 $\text{T}_{\text{E}}\text{X}$ 把段落压缩到最小的行数，但会损害其它审美上的考虑，比如避免过紧的单词间距。Plain $\text{T}_{\text{E}}\text{X}$ 设定 `\linepenalty` 为 10。

`\adjdemerits` [*$\langle number \rangle$* parameter]

此参数给行断点设定额外的缺陷（demerit），只要该断点在“视觉不相容的”相邻两行间出现。这些相邻行使得段落看起来参差不齐。不相容性取决于各行的松紧度⁶：

- 1) 此行是过紧的，如果其粘连至少需要收缩50%。
- 2) 此行是适中的，如果它的劣度小于或等于12。
- 3) 此行是松散的，如果其粘连至少需要伸长50%。
- 4) 此行是空荡的，如果其粘连需要伸长太多以致它的劣度超过100。

如果相邻的两行分类却不相近，比如，松散的行后面跟着一个过紧的行，或者空荡的行后面是一个适中的行，则称它们为视觉不相容的。

缺陷以劣度的平方为单位，因此只有在给定一个较大的数值时（比如数千）才会产生作用。Plain $\text{T}_{\text{E}}\text{X}$ 设定 `\adjdemerits` 为 10000。

`\exhyphenpenalty` [*$\langle number \rangle$* parameter]

此参数指定 $\text{T}_{\text{E}}\text{X}$ 在显式连字符（比如 ``helter-skelter'` 的连字符）处断行时所附加的惩罚值。增大此参数将阻碍 $\text{T}_{\text{E}}\text{X}$ 在显式连字符处换行。Plain $\text{T}_{\text{E}}\text{X}$ 设定 `\exhyphenpenalty` 为 50。

`\hyphenpenalty` [*$\langle number \rangle$* parameter]

此参数指定 $\text{T}_{\text{E}}\text{X}$ 在隐式连字符处断行时所附加的惩罚值。隐式连字符来自 $\text{T}_{\text{E}}\text{X}$ 的连字词典，或者来自你用 `\-`（第 ``@minus'` 页）插入的任意连字符。增大此参数将阻碍 $\text{T}_{\text{E}}\text{X}$ 将单词连字化。Plain $\text{T}_{\text{E}}\text{X}$ 设定 `\hyphenpenalty` 为 50。

⁶ 50%¹³

`\doublehyphendemerits` [*⟨number⟩* parameter]

此参数指定 T_EX 在导致连续两行都以连字符结尾的断点处所附加的额外缺陷值。增大此参数将阻碍 T_EX 将连续两行连字化。缺陷以劣度的平方为单位，因此只有在给定一个较大的数值时（比如数千）才会产生作用。Plain T_EX 设定 `\doublehyphendemerits` 为 10000。

`\finalhyphendemerits` [*⟨number⟩* parameter]

此参数指定 T_EX 在段落倒数第二行以连字符结尾时所附加的额外缺陷值。由于该行下面可能有短行造成的空白，这样的连字符通常认为是缺乏美感的。增大此参数将阻碍 T_EX 将段落倒数第二行连字化。缺陷以劣度的平方为单位，因此只有在给定一个较大的数值时（比如数千）才会产生作用。Plain T_EX 设定 `\finalhyphendemerits` 为 5000。

`\binoppenalty` [*⟨number⟩* parameter]

此参数指定 T_EX 在段内公式的二元运算符后断行所附加的惩罚值。Plain T_EX 设定 `\binoppenalty` 为 700。

`\relpenalty` [*⟨number⟩* parameter]

此参数指定 T_EX 在段内公式的二元关系符后断行所附加的惩罚值。Plain T_EX 设定 `\relpenalty` 为 500。

■

☆ \-

此命令在单词中加入“自定连字符”。自定连字符允许 T_EX 在该处连字化。但 T_EX 未必非得如此连字——它仅在必要时这样做。当某个单词在文档中出现一次或多次，而 T_EX 又找不到合适连字点时，这个命令就比较有用了。

```
Alice was exceedingly reluctant to shake hands first
with either Twee\-dle\-dum or Twee\-dle\-dee, for
fear of hurting the other one's feelings.
```


Alice was exceedingly reluctant to shake hands first with either Tweedledum or Tweedledee, for fear of hurting the other one's feelings.

`\discretionary { ⟨pre-break text⟩ } { ⟨post-break text⟩ } { ⟨no-break text⟩ }`

此命令指定一个“自定断点”，即， $\text{T}_{\text{E}}\text{X}$ 可以断行的位置。它还告诉 $\text{T}_{\text{E}}\text{X}$ 在断点前后放置的文字。

- 如果 \TeX 不在该处断行，它使用 $\langle no\text{-}break\text{ text}\rangle$ 。
- 如果 \TeX 确实在该处断行，它把 $\langle pre\text{-}break\text{ text}\rangle$ 放在断点前，而把 $\langle post\text{-}break\text{ text}\rangle$ 放在断点后。

如同 \-, 在自定断点处 T_EX 未必非得断行。实际上, \- 通常等同于 \discretionary{-}{ }{}。

TeX 有时也会加入自己的自定断点。例如，它在显式连字号或者破折号后加入 `\discretionary{}{}{}`。

```
% An ordinary discretionary hyphen (equivalent to \-):
\discretionary{-}{}{}

% A place where TeX can break a line, but should not
% insert a space if the line isn't broken there, e.g.,
% after a dash:
\discretionary{}{}{}

% Accounts for German usage: 'flicken', but 'flik-
% ken':
German "f\li\discretionary{k-}{k}{ck}en"
```

$$\backslash\mathrm{hyphenation}\ \{\langle word\rangle\sqcup\ldots\sqcup\langle word\rangle\}$$

$\text{T}_{\text{E}}\text{X}$ 中有个词典记录了它的连字规则的例外情形。词典中每个条目指明了某个单词应该如何连字化。这个 `\hyphenation` 命令用于添加单词到该词典。它的参数是用空格分开的多个单词，不区分字母大小写。单词中的连字符指明 $\text{T}_{\text{E}}\text{X}$ 可以在哪些地方将它连字化。一个不含连字符的单词将永远不会被连字化。然而，在特定单词中的 `\-` 还是优先于连字词典中对该单词的规定。你需要提供单词的各种语法形式给 $\text{T}_{\text{E}}\text{X}$ 处理，比如单数和复数形式。

```
\hyphenation{Gry-phon my-co-phagy}
\hyphenation{man-u-script man-u-scripts piz-za}
```

`\uchyph` [*<number>* parameter]

取 `\uchyph` (大写单词连字化) 为正数, 将允许对专有名词等以大写字母开始的单词连字化。取为零或者负数将禁止此类连字。Plain T_EX 设定 `\uchyph` 为 1, 因此 T_EX 一般会试着将大写字母开始的单词连字化。

`\showhyphens` {*<word>* □ ... □ *<word>*}

此命令在文档中一般不会用到, 使用它你可以在终端输出中看到 T_EX 对一些单词是如何连字的。用连字符表示的单词将出现在编译记录和终端输出中。同时你也会看到关于“松散的盒子”的警告, 忽略它即可。

```
\showhyphens{threshold quizzical draughts argumentative}
```

```
Underfull \hbox (badness 10000) detected at line 0
[] \tenrm thresh-old quizzi-cal draughts ar-gu-men-ta-tive
```

`\language` [*<number>* parameter]

不同的语言有不同的连字规则。此参数确定 T_EX 使用哪组连字规则。通过改变 `\language`, 你可以让 T_EX 对部分或整个文档用特定语言的连字规则连字。你所用的 T_EX 会说明 是否还有另外的连字规则可用 (除了英语), 以及适当的 `\language` 值是哪些。默认的 `\language` 值为 0。

在段落开始处, T_EX 设定当前语言值为 0。在添加字符到当前段落时, T_EX 会比较 `\language` 和当前语言值。如果两者不同, T_EX 添加一个小玩意, 表明当前语言的变化。这个小玩意提示后续处理时所用的语言规则应该改变。

`\setlanguage` *<number>*

此命令通过添加小玩意设定当前语言值为 *<number>*。该小玩意和修改 `\language` 时的一样。然而, 此命令不会改动 `\language` 的值。

```
\lefthyphenmin    [⟨number⟩ parameter]
\righthyphenmin   [⟨number⟩ parameter]
```

这两个参数指定，对于连字单词，T_EX 在连字符左右至少得有几个字母。Plain T_EX 中两者的默认值分别为 2 和 3；这是英语的建议值。

```
\hyphenchar ⟨font⟩ [⟨number⟩ parameter]
```

T_EX 未必总是以 '-' 字符为连字符。实际上，它使用的是当前字体的 `\hyphenchar`，此字符通常是 '-' 但未必总是。若某字体的 `\hyphenchar` 值为负数，T_EX 将不会对该字体下的单词连字化。

注意 *⟨font⟩* 指的是该字体的控制序列名称，而不是字体文件名 *⟨fontname⟩*。谨记：在编组结束时所分配的 `\hyphenchar` 撤销。若要局部改变 `\hyphenchar`，你必须显式地保存和恢复原有取值。

```
\hyphenchar\tenrm = '-'
% Set hyphenation for tenrm font to '-'.
\hyphenchar\tentt = -1
% Don't hyphenate words in font tentt.
```

```
\defaultshyphenchar [⟨number⟩ parameter]
```

当 T_EX 碰到 `\font` 命令时，它读取该字体的度量文件，并设置该字体的 `\hyphenchar` 为 `\defaultshyphenchar`。如果载入字体时 `\defaultshyphenchar` 的值不在 0--255 范围中，T_EX 将不会对该字体下的任何单词连字化；除非你以后用 `\hyphenchar` 命令设定此字体的连字符。Plain T_EX 设定 `\defaultshyphenchar` 为 45，即 '-' 字符的 ASCII 码。

```
\defaultshyphenchar = '-'
% Assume '-' is the hyphen, unless overridden.
\defaultshyphenchar = -1
% Don't hyphenate, unless overridden.
```

`\pretolerance` (第 `\pretolerance` 页)。

☆ `\beginsection <argument>\par`

此命令开始文档的一节。`<argument>` 用于表示节标题。标题将用粗体显示，靠左对齐，而且上下有额外的竖直间距。你也可以在 `<argument>` 后用一个空行得到 `\par`。

```
$\ldots$ till she had brought herself down to nine
inches high.
```

```
\beginsection Section 6. Pig and Pepper
```

```
For a minute or two she stood looking at the house $\ldots$■
```

```
... till she had brought herself down to nine inches high.
```

Section 6. Pig and Pepper

```
For a minute or two she stood looking at the house ...
```

`\item <argument>`

`\itemitem <argument>`

这两个命令用于创建逐项列表。在 `<argument>` 后面的整个段落会有 `\parindent` (对于 `\item`) 或者 `2\parindent` (对于 `\itemitem`) 的缩进。(`\parindent` 的解释见第 [`parindent'](#) 页。) `<argument>` 加上 `1en` 的间距后被放置在段落首行文字的左边。也就是说，它总是落在由 `\parindent` 确定的段落缩进空白中。

要让一个列表项包含多个段落，可以将 `\item{}` 放在另外的段落前。

```

{\parindent = 18pt
\noindent Here is what we require:
\item{1.}Three eggs in their shells,
but with the yolks removed.
\item{2.}Two separate glass cups containing:
\itemitem{(a)}One-half cup {\it used} motor oil.
\itemitem{(b)}One cup port wine, preferably French.
\item{3.}Juice and skin of one turnip.}

```

Here is what we require:

1. Three eggs in their shells, but with the yolks removed.
2. Two separate glass cups containing:
 - (a) One-half cup *used* motor oil.
 - (b) One cup port wine, preferably French.
3. Juice and skin of one turnip.

☆ `\proclaim <argument>.\□<general text>\par`

这个命令“陈述”一个定理、引理、假设等等。它用粗体显示 `<argument>`，并用斜体显示后面的段落。`<argument>` 后面必须跟着句号和空格，以此区分开 `<argument>` 和 `<general text>`。`<general text>` 到下一个段落之前截止。若要包含多个段落，你可以用 `\endgraf` 而不是空行或 `\par` 结束段落。

```

\proclaim Theorem 1.
What I say is not to be believed.

```

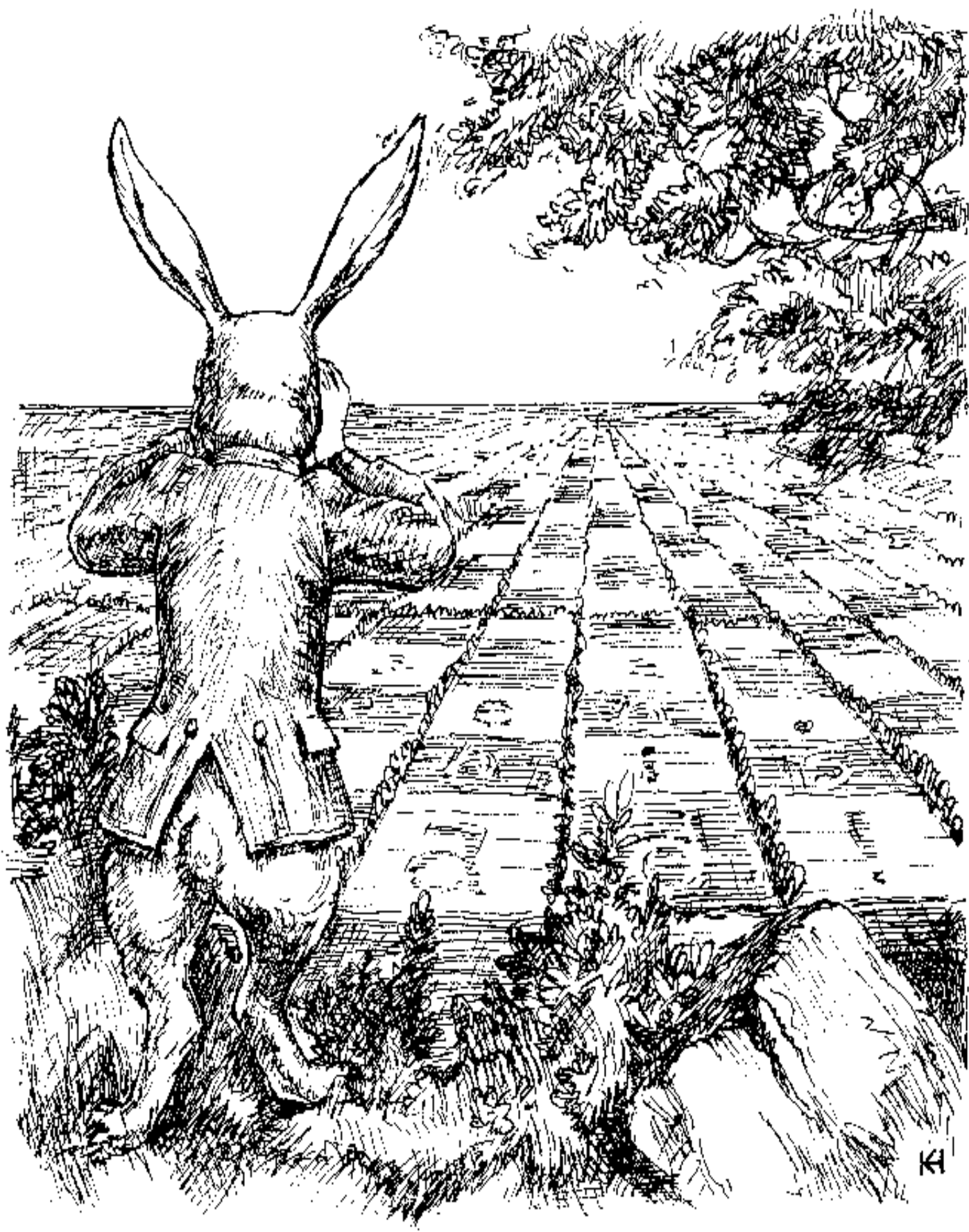
```

\proclaim Corollary 1. Theorem 1 is false.\par

```

Theorem 1. *What I say is not to be believed.*

Corollary 1. *Theorem 1 is false.*



6



`'cmddesc'`

```
\baselineskip    [ <glue> parameter ]
\lineskiplimit   [ <dimen> parameter ]
\lineskip        [ <glue> parameter ]

TeX
TeX \baselineskip \baselineskip \prevdepth \lineskiplimit
TeX \lineskip1 The TeXbook 79–80
\baselineskip \lineskip The TeXbook 78 \lineskiplimit
\baselineskip \baselineskip
```

¹ `TeX \prevdepth -1000 pt \prevdepth \prevdepth ≤ -1000 pt`


```
\baselineskip = 11pt \lineskiplimit = 1pt
\lineskip = 2pt plus .5pt
```

Sometimes you'll need to typeset a paragraph that has tall material, such as a mathematical formula, embedded within it. An example of such a formula is $\$n \choose k\$$. Note the extra space above and below this line as compared with the other lines. (If the formula didn't project below the line, we'd only get extra space above the line.)

Sometimes you'll need to typeset a paragraph that has tall material, such as a mathematical formula, embedded within it. An example of such a formula is $\binom{n}{k}$. Note the extra space above and below this line as compared with the other lines. (If the formula didn't project below the line, we'd only get extra space above the line.)

```
\baselineskip = 2\baselineskip % Start double spacing.
```

```
\prevdepth [⟨dimen⟩ parameter]
```

```
TeX \prevdepth TeX \prevdepth -1000pt
```

```
\normalbaselineskip [⟨glue⟩ parameter]
```

```
\normallineskiplimit [⟨dimen⟩ parameter]
```

```
\normallineskip [⟨glue⟩ parameter]
```

```
\normalbaselines
```

```
\baselineskip\lineskip \lineskiplimit \normalbaselines
```

```
\baselineskip\lineskip \lineskiplimit
```

```
\offinterlineskip
```

TeX struts), 而不影响 TeX 的正常行间粘连。在构造水平阵列时经常会用到 `\offinterlineskip`。

```

\def\entry#1:#2 {\strut\quad#1\quad&\quad#2\quad\cr}
\offinterlineskip \tabskip = 0pt \halign{%
\vrule\quad\hfil#\hfil\quad\vrule&
\quad\hfil#\hfil\quad\vrule\cr
\noalign{\hrule}
\vphantom{\vrule height 2pt}&\cr \noalign{\hrule}
\entry \it Opera:\it Composer
\vphantom{\vrule height 2pt}&\cr \noalign{\hrule}
\vphantom{\vrule height 2pt}&\cr
\entry Fidelio:Beethoven
\entry Peter Grimes:Britten
\entry Don Giovanni:Mozart
\vphantom{\vrule height 2pt}&\cr \noalign{\hrule}}

```

<i>Opera</i>	<i>Composer</i>
Fidelio	Beethoven
Peter Grimes	Britten
Don Giovanni	Mozart

```
\nointerlineskip
```

TeX

```
\openup <dimen>
```

```

\baselineskip <dimen> \openup \baselineskip \openup
\openup

```

Alice picked up the White King very gently, and lifted him across more slowly than she had lifted the Queen; but before she put him on the table, she thought she might well dust him a little, he was so covered with ashes.

```
\openup .5\baselineskip % 1.5 linespacing.
```

Alice picked up the White King very gently, and lifted him across more slowly than she had lifted the Queen; but before she put him on the table, she thought she might well dust him a little, he was so covered with ashes.



`\break`

vbox。 `\break` 也能用在水平模式中。

`\nobreak`

`\nobreak`

`\allowbreak`

TeX `\allowbreak`

`\penalty` *number*

penalty) 项。惩罚项使得 TeX 或多或少愿意在此处分页。负惩罚值, 实际上是奖励值, 鼓励分页; 正惩罚值阻碍分页。大于或等于 10000 的惩罚值彻底阻止分页, 而小于或等于 -10000 的惩罚值强制分页。此命令也能用于水平模式。

```
\def\break{\penalty-10000 } % as in plain TeX
```

```
\def\nobreak{\penalty10000 } % as in plain TeX
```

```
\def\allowbreak{\penalty0 } % as in plain TeX
```

`\goodbreak`

TeX

`\smallbreak`

`\medbreak`

`\bigbreak`

`TEX \smallskip\medskip\bigskip ‘smallskip’ TEX`

☆ `\eject`

`\supereject`

`\vfil` (第 `\vfil` 页), `TEX` 将试着将伸展页面内容 (很可能会警告有松散的竖直盒子)。 `\supereject` 命令还让 plain `TEX` 输出例行程序处理剩下的插入项, 例如长脚注, 在处理其他输入之前生成它们。因而, `\supereject` 更适合在文档各章或主要单元结束时使用。

`\filbreak`

`TEX \filbreak \filbreak TEX \filbreak TEX ‘filbreak’`

`\raggedbottom`

`\normalbottom`

`TEX \raggedbottom TEX \raggedbottom \normalbottom`
`\raggedbottom`

■

`\interlinepenalty` [*⟨number⟩* parameter]

10000 `TEX Plain TEX \interlinepenalty 0`

`\clubpenalty` [*⟨number⟩* parameter]

“孤行”。 Plain `TEX` 设定 `\clubpenalty` 为 150。

`\widowpenalty` [*⟨number⟩* parameter]

“寡行”。 Plain `TEX` 设定 `\widowpenalty` 为 150。

`\displaywidowpenalty` [*⟨number⟩* parameter]

Plain \TeX \displaywidowpenalty 50.

\displaywidowpenalty [$\langle number \rangle$ parameter]
 Plain \TeX
 \displaywidowpenalty 10000

\displaywidowpenalty [$\langle number \rangle$ parameter]
 Plain \TeX
 \displaywidowpenalty 0

\displaywidowpenalty [$\langle number \rangle$ parameter]
 \displaywidowpenalty \hyphenpenalty ‘ \hyphenpenalty ’ Plain \TeX
 \displaywidowpenalty 100.

\displaywidowpenalty [$\langle number \rangle$ parameter]
 \TeX \TeX \displaywidowpenalty *The \TeX book* 123–125
 \displaywidowpenalty *The \TeX book* 125

\displaywidowpenalty [$\langle number \rangle$ parameter]
 \TeX \displaywidowpenalty \displaywidowpenalty Plain \TeX \displaywidowpenalty ■
 0

\displaywidowpenalty [$\langle dimen \rangle$ parameter]
 \TeX \displaywidowpenalty \displaywidowpenalty \displaywidowpenalty \TeX \TeX \TeX
 \displaywidowpenalty \displaywidowpenalty

\displaywidowpenalty [$\langle dimen \rangle$ parameter]
 \TeX \displaywidowpenalty

\displaywidowpenalty [$\langle dimen \rangle$ parameter]
 \TeX \displaywidowpenalty

\displaywidowpenalty [$\langle dimen \rangle$ parameter]
 \TeX \displaywidowpenalty

```

\pagestretch    [  $\langle dimen \rangle$  parameter ]
\pagefilstretch  [  $\langle dimen \rangle$  parameter ]
\pagefillstretch [  $\langle dimen \rangle$  parameter ]
\pagefilllstretch [  $\langle dimen \rangle$  parameter ]
 $n_0 + n_1 \text{fil} + n_2 \text{fill} + n_3 \text{filll}$   $n_i$  TEX

```



```

\hsize    [  $\langle dimen \rangle$  parameter ]
Plain TEX \hsize 6.5in ‘hsize’

```

```

\vsizer    [  $\langle dimen \rangle$  parameter ]
TEX \vsizer \pagegoal ‘pagegoal’ \vsizer \pagegoal Plain
TEX \vsizer 8.9in

```

```

\hoffset    [  $\langle dimen \rangle$  parameter ]
\voffset    [  $\langle dimen \rangle$  parameter ]
TEX2 \hoffset \voffset \hoffset \voffset TEX

```

```

\hoffset = -.3in
% Start printing .7 inches from left edge of paper.
\voffset = 1in
% Start printing 2 inches from top edge of paper.

```

```

\topskip    [  $\langle glue \rangle$  parameter ]
 $d$  TEX \topskip  $d$   $d$  \topskip  $d$  TEX \topskip
\topskip \topskip \topskip \topskip  $e$  \topskip +  $e$  The
TEXbook 113–114 \topskip Plain TEX \topskip 10pt

```

² T_EX .dvi

`\parskip` [*<glue>* parameter]
`\TeX \par` (第 `\@par` 页) `\TeX Plain \TeX \parskip 0pt`
plus 0.1pt

`\maxdepth` [*<dimen>* parameter]
`\boxmaxdepth 'boxmaxdepth' \maxdepth \TeX \maxdepth Plain-`
`\TeX \maxdepth 4pt`



`\pageno` [*<number>* parameter]

front matter) 中页码通常是负数, 我们用小写罗马数字而不用阿拉伯数字编号。在页面中间对页码的改动, 会在该页的页眉和页脚中用到。页码事实上由 `\TeX` 的输出例行程序打印, 你可以修改它们。

`Plain \TeX` 将页码记录在寄存器 `\count0` 中。(`\pageno` 实际上就是 `\count0` 的别名。) 当它输出一个页面到 `.dvi` 文件时, `\TeX` 在终端中显示 `\count0` 的值, 让你知道它在排版哪一页。也可以把寄存器 `\count1--\count9` 用于嵌套的多级页码 (但你必须自己编程)。如果这些寄存器的值非零, `\TeX`³

This explanation appears on page `\number\pageno`
of our book.

This explanation appears on page 135 of our book.

`\pageno = 30` % Number the next page as 30.
Don't look for this explanation on page `\number\pageno`.

Don't look for this explanation on page 30.

`\advancepageno`
 $n \geq 0 \quad n = 1 \quad n < 0 \quad n = 1$

³ `\count0 \count9 \count0 \count3 (17,0,0,7) 0 \TeX [17.0.0.7]`

☆ \nopagenumbers

plain T_EX T_EX

\folio

\pageno n $n \geq 0$ T_EX n $n < 0$ T_EX $-n$

This explanation appears on page \folio\ of the book.

This explanation appears on page 136 of the book.

■

\headline [*<token list>* parameter]

\footline [*<token list>* parameter]

Plain T_EX

\hsize \hfil 'hfil' T_EX T_EX

\headline \footline T_EX 'bighead'

\headline = {\tenrm My First Reader\hfil Page \folio}

My First Reader

Page 10

(*at the top of page 10*)

\footline = {\tenit\ifodd\pageno\hfil\folio

\else\folio\hfil\fi}

% Produce the page number in ten-point italic at

% the outside bottom corner of each page.

■

\mark { *<text>* }

T_EX *<mark text>* \hbox\vbox \vtop T_EX \mark hbox \mark

The T_EXbook 259–260 \mark


```

\firstmark
\botmark
\topmark
\mark TEX \box255 TEX

```

- \firstmark
- \botmark
- \topmark \botmark

```

TEX \firstmark \botmark \topmark The TEXbook 258 \firstmark\botmark
\topmark

```

```

\splitfirstmark
\splitbotmark
\mark V \vsplit 'vsplit' V TEX
▪ \splitfirstmark V
▪ \splitbotmark V
\vsplit \vsplit

```

■

☆ \footnote $\langle argument_1 \rangle$ $\langle argument_2 \rangle$
 \vfootnote $\langle argument_1 \rangle$ $\langle argument_2 \rangle$
 $\langle argument_1 \rangle$ $\langle argument_2 \rangle$
 \footnote \vfootnote

\vfootnote \footnote \vfootnote \vfootnote \vfootnote
\vfootnote

To quote the mathematician P\'olya is a ploy.\footnote
*{This is an example of an anagram, but not a strict one.}

To quote the mathematician Pólya is a ploy.*

⋮

* This is an example of an anagram, but not a strict one.

\$\$\$f(t)=\sigma\sigma t\;;\;\raise 1ex \hbox{\dag}\$\$\$
\vfootnote \dag{The $\sigma\sigma$ notation was explained
in
the previous section.}

$$f(t) = \sigma\sigma t^\dagger$$

⋮

† The $\sigma\sigma$ notation was explained in the previous section.

■

\topinsert <vertical mode material> \endinsert
\midinsert <vertical mode material> \endinsert
\pageinsert <vertical mode material> \endinsert

TEX <vertical mode material>

- \topinsert \topinsert
- \midinsert \midinsert
- \pageinsert \vfil

TEX <vertical mode material>

TeX

```
\pageinsert
% This text will appear on the following page, by itself.
This page is reserved for a picture of the Queen of Hearts
sharing a plate of oysters with the Walrus and
the Carpenter.
\endinsert
```

```
\endinsert
\topinsert\midinsert \pageinsert
```

```
\insert <number> { <vertical mode material> }
\footnote\vfootnote\topinsert\midinsert \pageinsert \insert
```

```
4 n \newinsert '@newinsert' \insert <vertical mode material>
\box n
```

TeX

- \box n TeX n TeX \box n n \ifvoid '@ifvoid' \box n TeX
- \count n f TeX nf/1000 f 500 0
- \dimen n n
- \skip n n TeX

TeX \box n TeX The TeXbook 122–125 TeX

\floatingpenalty (第 \floatingpenalty' 页).

\output [<token list> parameter]

TeX TeX \box255 \output \box255 dvi

⁴ The TeXbook ‘

`\plainoutput`

`plain TEX Plain TEX \output \plainoutput`

`\shipout <box>`

`TEX <box> .dvi. TEX \shipout <box> \write \shipout`

`\deadcycles` [<number> parameter]

`TEX \shipout 5 \deadcycles TEX`

`\maxdeadcycles` [<number> parameter]

`\deadcycles \maxdeadcycles TEX TEX \shipout\box255 Plain
TEX \maxdeadcycles 25`

`\outputpenalty` [<number> parameter]

`TEX TEX \outputpenalty \outputpenalty 0
TEX TEX \penalty\outputpenalty`

`\holdinginserts` [<number> parameter]

`TEX 0 TEX \vsize (第 'vsize' 页) 1`

`\vsplit <number> to <dimen>`

`TEX <number> B2 B2 B2 TEX B1 B2 \vsplit B1 B1
B2 B2 \vsplit
TEX <dimem> B1 \pagegoal B1 <dimem> TEX TEX B2`

`\setbox 20 = \vsplit 30 to 7in`

`=`

`% 30 号盒子截去 7 英寸，并放到 20 号盒子中。`

`\splitmaxdepth` [<dimen> parameter]

`\vsplit \splitmaxdepth \maxdepth (第 'maxdepth' 页)`

⁵ `TEX \shipout \deadcycles 0 \output 1`

`\splittopskip` [*⟨glue⟩* parameter]

`\TeX \vsplit \splittopskip \topskip` (第 `\topskip` 页)

`\splitbotmark, \splitfirstmark` (第 `\splitfirstmark` 页).





7



`'cmddesc'`



`\thinspace`

1/6 em (第 `'dimdefs'` 页), 也就是说, 它使得 $\text{T}_{\text{E}}\text{X}$ 向右移动一个相应的距离。例如, 当你需要嵌套引用, 并希望将不同层次的引号分开时, 它便非常有用。 $\text{T}_{\text{E}}\text{X}$ `\thinspace`

`“‘\thinspace‘一个引用。’\thinspace’’\par`
`24,\thinspace 29--31,\thinspace 45,\thinspace 102`

`“‘一个引用。’”`
`24, 29–31, 45, 102`

`\negthinspace`

1/6 em (第 `'dimdefs'` 页), 也就是说, 它使得 `TEX` 向左移动一个相应的距离。当需要将两个间隔稍远的字符靠近一些时便非常有用。
`TEX \negthinspace`

The horror, the horror\negthinspace, the horror of it all!

The horror, the horror; the horror of it all!

`\enspace`

1 en 1 em `'dimdefs'` `TEX \enspace \enspace`

1.\enspace

1.

☆ `\enskip`

`\quad`

`\qquad`

`TEX plain TEX cmr10`

<code>\enskip</code>	1/2 em	→ ←
<code>\quad</code>	1 em	→ ←
<code>\qquad</code>	2 em	→ ←

en\enskip skip; quad\quad skip; qquad\qquad skip

en skip; quad skip; qquad skip

■

☆ \smallskip

\medskip

\bigskip

<u>smallskip</u>	<u>medskip</u>	<u>bigskip</u>
<u> </u>	<u> </u>	<u> </u>

\smallskip 3 1 \medskip \smallskip\bigskip \medskip

Hop \smallskip skip \medskip and \bigskip jump.

Hop

skip

and

jump.

\smallskipamount [*⟨glue⟩* parameter]

\medskipamount [*⟨glue⟩* parameter]

\bigskipamount [*⟨glue⟩* parameter]

\smallskip\medskip \bigskip plain T_EX 1/4 1/2 1 \baselineskip ‘*baselineskip*’ ■

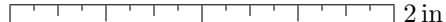


☆ `\hskip <dimen1> plus <dimen2> minus <dimen3>`
`\vskip <dimen1> plus <dimen2> minus <dimen3>`
`<dimen1> \hskip <dimen1> \vskip <dimen1> <dimen1> <dimen2>`
`<dimen3> plus <dimen2> minus <dimen3> plus minus <dimen>`
`\hskip mu “ mu \mskip ‘mskip’`

`\hbox to 2in{one\hskip 0pt plus .5in two}`

one two
 2 in

`\hbox to 2in{Help me! I can't fit`
`{\hskip 0pt minus 2in} inside this box!}`

Help me! I can't fit inside this box!
 2 in

`\vbox to 4pc{\offinterlineskip% \vskip`
`\hbox{-}\vskip 0pc plus 1pc \hbox{二}`
`\vskip .5pc \hbox{三}}`

`\hglue <glue>`
`\vglue <glue>`
`\hglue \vglue \hskip \vskip \vglue \topglue`

```
\topglue <glue>
1 \hoffset \voffset \topskip
TeX \topskip \topglue
```

```
\kern <dimen>
```

```
TeX
```

```
▪ TeX
```

```
▪ TeX
```

```
TeX TeX
```

```
(a) (b) TeX TeX \hglue \vglue
```

```
\kern mu <dimen> ‘ mu \mkern ‘mkern’
```

```
\centerline{${\Downarrow}$}\kern 3pt %
\centerline{${\Longrightarrow}$}\kern 6pt %
{\bf } \kern 6pt %
${\Longleftarrow}$}
\kern 3pt %
\centerline{${\Uparrow}$}
```

```

      ↓↓
⇒      ⇐
      ↑↑
```

```
\hfil \vfil
\hfill \vfill
```

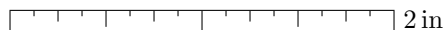
```
\hfil \hfill \vfil \vfill
\vfil \vfill
```

```
\hbox to 2in{左 \hfil \hfil }
```

```
|_|_|_|_|_|_|_|_|_|_|_|_|_|_| 2 in
```

¹ \topglue TeX 3.0 TeX ‘xintex’ The TeXbook 18th

```
\hbox to 2in{左 \hfil \hfill }
```



```
| 2in
```

```
\leftline{%
\vbox to 4pc{%
  \hbox{上}\vfil\hbox{中}\vfil \hbox{下}}\quad
\vbox to 4pc{%
  \hbox{上}\vfil\hbox{中}\vfill\hbox{下}}}
```

```
\hss
```

```
\vss
```

```
\hss \vss
```

```
\line{正文文本 \hfil\hbox to Opt{页边空白 \hss}}
% ‘页边空白 \hss’
```

```
\vbox to 1pc{\hrule width 6pc % Top of box.
  \hbox{1} \vskip 1pc\hbox to 2pc{\hfil 2}
  % \vss \vskip
  \vss \hbox to 3pc{\hfil 3}
  \hrule width 6pc}%
```

3

```

\hfilneg
\vfilneg
    \hfil \vfil \hfil \vfil \hfilneg \vfilneg n \hfilneg n
\hfil \vfilneg \hfilneg \vfilneg \hfil \vfil
    \hfilneg \vfilneg \hfil \vfil

\leftline{\hfil \hfilneg} =
% \leftline \hfil。

\def\aa{\hbox to 1pc{\hfil 2}\vfil}
\vbox to 4pc{\hbox{1} \vfil \aa
    \vfilneg \hbox to 2pc{\hfil 3}}

1

2
3

\hbadness \vbadness ‘vbadness’ \hfuzz \vfuzz ‘vfuzz’ ‘

```

■ hbox vbox

```

\hbox { <horizontal mode material> }
\hbox to <dimen> { <horizontal mode material> }
\hbox spread <dimen> { <horizontal mode material> }

```

hbox（水平盒子）以容纳 *<horizontal mode material>*。*<horizontal mode material>* 两侧的大括号定义一个编组。T_EX 不会对 *<horizontal mode material>* 断行，因为当其组装箱子时它处于受限水平模式下。一旦盒子生成后，T_EX

```
\hbox \hbox TEX TEX hbox
```

`hbox \hbox`

- `<horizontal mode material> hbox`
- `to <dimen> hbox <dimen>`
- `spread <dimen> hbox <dimen> hbox <dimen>`

`hbox \hfil 'hfil'`

```
\hbox{ugly suburban sprawl}
\hbox to 2in{ugly \hfil suburban \hfil sprawl}
\hbox spread 1in {ugly \hfil suburban \hfil sprawl}
% \hfil, 将会得到 `underfull hbox' 警告。
```

```
ugly suburban sprawl
ugly      suburban      sprawl
ugly      suburban      sprawl
|-----| 3 in
```

```
\vtop <vertical mode material>
\vtop to <dimen> { <vertical mode material> }
\vtop spread <dimen> { <vertical mode material> }
\vbox { <vertical mode material> }
\vbox to <dimen> { <vertical mode material> }
\vbox spread <dimen> { <vertical mode material> }
```

`vbox`（竖直盒子）以容纳 `<vertical mode material>`。`<vertical mode material>` 两侧的大括号定义一个编组。当 $\text{T}_{\text{E}}\text{X}$ 组装盒子时它处于内部竖直模式下。一旦盒子生成后， $\text{T}_{\text{E}}\text{X}$

`\vtop \vbox` $\text{T}_{\text{E}}\text{X}$ `vbox \vtop vbox \vbox vbox \vtop vbox`，其顶部将会近似在一条线上；相应地，一行全部由 `\vbox` 构成的 `vbox`，其底部将会近似在一条线上。

当你希望将一些文字放在单独的一页上时，`\vtop` 和 `\vbox` 是有用的。（为了这个目的，它通常不在乎你使用的命令。）如果你使用这些命令使得 $\text{T}_{\text{E}}\text{X}$ 不能以一种可以接受的方式分页时， $\text{T}_{\text{E}}\text{X}$ 会在 `\output` 处于活动的情况下，给出 `vbox` 过满或不足的警告。

`vbox` 的高度取决于 `\vtop` 或 `\vbox` 的参数。对于 `\vbox`， $\text{T}_{\text{E}}\text{X}$

- `<vertical mode material> vbox`
- `to <dimen> vbox <dimen>`
- `spread <dimen> vbox <dimen> vbox <dimen>`

```
\vtopTeX \vbox
  vbox2
```

- \vtop
 - \vbox³
- ```
vbox \vfil 'vfil'
```

```
\hbox{\hsize = 10pc \raggedright\parindent = 1em
```

\vtop{在这个例子中，我们将看到如何使用 vbox 产生双栏的效果。

每个 vbox 都包含两个段落，除了设置为左对齐以外均依照 \TeX 通常的规则进行排版。 \par

```
}
\hskip 2pc
\vtop{\noindent
```

```
\par
 vbox
```

```
}}
```

```
vbox vbox TEX
```

```
vbox
```

```
\hbox{\hsize = 1in \raggedright\parindent = 0pt
```

```
\vtop to .75in{\hrule .75in。 \vfil\hrule}
```

```
\qqquad
```

```
\vtop{\hrule 这个盒子具体其本身的深度。 \vfil\hrule}
```

```
\qqquad
```

```
\vtop spread .2in{\hrule 这个盒子的深度比其自身的深度多 .2in。
```

```
\vfil\hrule}}
```

---

<sup>2</sup> vbox \moveright \moveleft vbox

<sup>3</sup> \boxmaxdepth \boxmaxdepth



---

.75in。

---



---

.2in。

---

```
% \vbox
\hbox{\hsize = 1in \raggedright
\vbox to .5in{\hrule .5in. \vfil\hrule}
\quad
\vbox to .75in{\hrule .75in. \vfil\hrule}}
```

---

.5in。

---

.75in。

`\boxmaxdepth` [ *<dimen>* parameter ]

$D$   $D_{\text{TeX}}$   $d$   $D_{\text{TeX}}$   $d - D$  `\boxmaxdepth`  $\text{TeX}$  Plain  $\text{TeX}$   
`\boxmaxdepth` `\maxdimen` ‘`maxdimen`’ `\boxmaxdepth`

`\underbar` *<argument>*

*<argument>* hbox，并在其下划线，而不考虑任何突出到盒子的基线以下的东西。

`\underbar`{为什么不学习 `\TeX`?}

=

Why not learn  $\text{TeX}$ ?

`\everyhbox` [ *<token list>* parameter ]

`\everyvbox` [ *<token list>* parameter ]

$\text{TeX}$  hbox vbox

■

```

\setbox <register> = <box>
\box <register>
 <register> \setbox n = \box n =
 \copy \box \copy TEX

```

```

\setbox0 = \hbox{蘑菇}
\setbox1 = \vbox{\copy0\box0\box0}
\box1

```

```

\copy <register>
 <register> \box

```

```

\setbox0 = \hbox{美好}
\copy0 \box0 \box0

```

```

\unhbox <register>
\unvbox <register>
 <register> \unhbox hbox \unvbox vbox \unhcopy \unvcopy TEX

```

```

\setbox0=\hbox{素甲鱼深深地叹息着，用一只手背抹着眼泪。}
\setbox1=\hbox{他想说话，可是有好一阵子泣不成声。}
\unhbox0 \unhbox1
% \box0 \box1 hbox
%
\box1 %

```

```
\unhcopy <register>
\unvcopy <register>
 <register> \unhcopy hbox \unvcopy vbox
```

```
\setbox0=\hbox{素甲鱼深深地叹息着，用一只手背抹着眼泪。} =
\setbox1=\hbox{他想说话，可是有好一阵子泣不成声。}
\unhcopy0 \unhcopy1\par\noindent
% \box0 \box1 hbox
%
\box1 % hbox。
```

```
\wd, \dp, \ht ‘ht’
```

■

```
\moveleft <dimen> <box>
\moveright <dimen> <box>
 <box> <dimen> \moveleft \moveright
```

```
\vbox{\vbox{Phoebe}\vbox{walked}}%
\moveleft 20pt\vbox{a}\moveright 20pt\vbox{crooked}%
\vbox{mile.}}
```

```
Phoebe
walked
```

a

```
crooked
mile.
```

```
\lower <dimen> <box>
\raise <dimen> <box>
 <box> <dimen> \raise \lower
```

\lower 6pt \hbox{肿块}, 你觉得 =  
           \raise 6pt \hbox{沮丧} 吗?

\produces %Are you feeling \lower 6pt \hbox{depressed} about  
 the % \raise 6pt \hbox{bump} on your nose?   \lower 6pt \hbox{肿  
 块}, 你觉得 \raise 6pt \hbox{沮丧} 吗? \endexample  
 \einx^^{盒子//移动盒子} \enddesc

%=====

%\subsection {Dimensions of box registers} \subsection {盒子寄存  
 器的尺寸}

%\begindesc %\bix^^{box registers} %\bix^^{dimensions//of box  
 registers} %\cts ht {\<register>\param{dimen}} %\cts dp {\<register>\param{dimen}}  
 %\cts wd {\<register>\param{dimen}} %\explain %These parameters  
 refer to the ^{height}, ^{depth}, and ^{width} %respectively of  
 \minref{box} \minref{register} \<register>. %You can use them  
 to find out the dimensions of a box. %You can also change the  
 dimensions of a box, but it's a tricky business; %if you want to be  
 adventurous you can %learn all about it from \knuth{pages~388--389}.

\begindesc \bix^^{盒子寄存器} \bix^^{尺寸//盒子寄存器的尺寸}  
 \cts ht {\<register>\param{dimen}} \cts dp {\<register>\param{dimen}}  
 \cts wd {\<register>\param{dimen}} \explain 这些参数分别表示  
 \minref{盒子}\minref{寄存器} \<register> 的 ^{高度}、^{深度}  
 和 ^{宽度}。你可以使用这些命令获得一个盒子的尺寸。你也  
 可以改变一个盒子的尺寸，但这是一个需要技巧的事情；如果  
 你希望成为冒险者，你可以从 \knuth{pages~388--389} 中获得全  
 部的技巧。 \example \setbox0 = \vtop{\hbox{一个}\hbox{米黄色  
 的}\hbox{兔子}}% 那个盒子的宽度是 \the\wd0, 高度是 \the\ht0,  
 深度是 \the\dp0。

0.0pt, 高度是 0.0pt, 深度是 28.0pt。

■

\strut

(8.5pt) (3.5pt) plain T<sub>E</sub>X cmr10 \offinterlineskip T<sub>E</sub>X

\strut

```

cmr10 \strut

\noindent %
\offinterlineskip %
% vbox
\top{\hbox{.}\hbox{.}\hbox{.x}
\hbox{.\vrule height 4pt depth 0pt}}\quad
%
% 下面这些 vbox 中的句号在竖直方向上则是等间距的。
\top{\hbox{.\strut}\hbox{.(\strut)}\hbox{.x\strut}
\hbox{.\vrule height 4pt depth 0pt\strut}}

: (.
: X . (
: .X
: .

```

`\mathstrut`

`\mathstrut` ‘`\vphantom`’。它主要用于使得根号、下横线和上横线能与公式中其它的根号、下横线和上横线成行排列。除能根据出现在数学公式中的不同样式自动调节以外，它与 `\strut`（第 `\strut` 页）非常相似。

```


$$\overline{a_1 a_2} \wedge \overline{b_1 b_2} \quad \text{versus} \quad \overline{a_1 a_2} \wedge \overline{b_1 b_2}$$

$$\sqrt{\epsilon} + \sqrt{\xi} \quad \text{versus} \quad \sqrt{\epsilon} + \sqrt{\xi}$$


```

$$\overline{a_1 a_2} \wedge \overline{b_1 b_2} \quad \text{versus} \quad \overline{a_1 a_2} \wedge \overline{b_1 b_2}$$

$$\sqrt{\epsilon} + \sqrt{\xi} \quad \text{versus} \quad \sqrt{\epsilon} + \sqrt{\xi}$$

`\phantom` *⟨argument⟩*

$\langle argument \rangle \phantom{\quad}$

$\$1\phantom{9}2\$$

$1\ 2$

$\hphantom{\langle argument \rangle}$

$\vphantom{\langle argument \rangle}$

- $\hphantom{\langle argument \rangle}$
- $\vphantom{\langle argument \rangle}$

$\left[\vphantom{\over u}t\right] \star \left[\over v\right] \quad \left[\over u\right] \quad \left[\vphantom{\over u}t\right]$

$$\begin{bmatrix} t \end{bmatrix} \star \begin{bmatrix} u \\ v \end{bmatrix} \quad \{ \quad \}$$

$\smash{\langle argument \rangle}$

$\langle argument \rangle \smash{\phantom{\quad}} \phantom{\quad}$

$\{\smash{r_m \brace r_n} \vphantom{r}\} \rightarrow r$

$$\left\{ \begin{matrix} r_m \\ r_n \end{matrix} \right\} \Rightarrow r$$

$\null$

$\hbox{.}$

$\setbox0 = \null$

$\null \the\wd0 \the\ht0$

$\the\dp0$

$0.0pt\ 0.0pt\ 0.0pt$



```
\overfullrule [⟨dimen⟩ parameter]
TeX Plain TeX 5pt
```

```
\hbadness [⟨number⟩ parameter]
\vbadness [⟨number⟩ parameter]
```

`\hbadness` `hbox`，而 `\vbadness` 适用于 `vbox`。如果一个盒子的恶劣状态超过了阈值，TeX plain TeX 1000TeX `\hbadness` `\vbadness` *The TeXbook* page 302，以获得 TeX 如何确定在什么时候对过满或未满盒子进行警告的准确描述。

```
\hbadness = 10000 % hbadness
\hbox to 2in{a b}\par
\vbadness = 500 % hbadness 500
\hbox to 2in{a\hskip 0pt plus .5in b}
```

```
Underfull \hbox (badness 5091) detected at line 4
\tenrm a b
```


```
\hbox(6.94444+0.0)x144.54, glue set 3.70787
.\tenrm a
.\glue 0.0 plus 36.135
.\tenrm b
```

```
\badness
TeX \badness 1000000 0 10000
```

```
\hfuzz [⟨dimen⟩ parameter]
\vfuzz [⟨dimen⟩ parameter]
TeX \hfuzz \vfuzz Plain TeX 0.1pt
```

```
\hfuzz = .5in
\hbox to 2in{This box is longer than two inches.}
%
```

This box is longer than two inches.



```
\tolerance ‘tolerance’
```

```
\lastkern
\lastskip
\lastpenalty
\lastbox
 \lastbox \lastkern 0
 \lastbox \lastbox \lastbox
```

```
\def\{二 \kern 15pt}
\{二\{二\hskip 2\lastkern \par
% ‘三’ 之前产生三倍的紧排。
\def\{二\{二\hskip 2\lastkern \par
- \{二
\setbox0 = \lastbox % ‘二’ 去掉。
三 \box0。
```

```
\unkern
\unskip
\unpenalty
 \lastbox TEX \unbox \lastbox
```



```

\hrule
\hrule height <dimen> width <dimen> depth <dimen>
\vrule
\vrule width <dimen> height <dimen> depth <dimen>

```

`\hrule \vrule`  $\TeX$  ---如果你这么做，起作用的是最后一个值。

如果你没有设定水平标线的宽度，则标线将水平地扩展到包含标线的盒子或阵列的最内侧的边界。如果你没有设定水平标线的高度，其默认值是 `0.4pt`；如果你没有设定水平标线的深度，其默认值是 `0pt`。

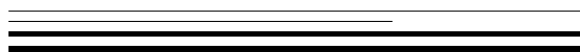
如果你没有设定竖直标线的宽度，其默认值是 `0.4pt`；如果你没有设定竖直标线的高度或深度，则标线将地扩展到包含标线的盒子或阵列的最内侧的边界。

$\TeX$  以内在的竖直项目处理水平标线，而以内在的水平项目处理竖直标线。因此，水平标线只有竖直模式下是有效的，而竖直标线只在水平模式下是有效的。如果觉得意外，可以直观化地观测。水平标线从左向右移动，并且将竖直的项目有序地分开，而竖直标线从上到下，将水平项目有序地分开。

```

\hrule\smallskip
\hrule width 2in \smallskip
\hrule width 3in height 2pt \smallskip
\hrule width 3in depth 2pt

```



```
% \hrule
\leftline{
 \vbox{\hrule width .6in height 5pt depth 0pt}
 \vbox{\hrule width .6in height 0pt depth 8pt}
 \vbox{\hrule width .6in height 5pt depth 8pt}
 \vbox{\hbox{基线}\kern 3pt \hrule width .6in}
}
```



```
\hbox{({\vrule} {\vrule width 8pt})}
\hbox {({\vrule height 13pt depth 0pt}
 {\vrule height 13pt depth 7pt} x)}
%
% 'x'
```

```
(| ■)
(| | x)
```

☆ `\leaders` *<box or rule>* *<skip command>*  
`\cleaders` *<box or rule>* *<skip command>*  
`\xleaders` *<box or rule>* *<skip command>*  
       ‘ ‘ *<box>* *<rule>* *<skip command>* *<skip command>*  $\mathrm{T_{E}X}$  *<skip command>*  $\mathrm{T_{E}X}$

- `\leaders $\mathrm{T_{E}X}$  \leaders`  $B$   $B$
- `\cleaders`
- `\xleaders`  $l$   $n$   $\mathrm{T_{E}X}$   $l/(n+1)$

```

\def\pattern{\hbox to 15pt{\hfil.\hfil}}
\line{Down the Rabbit-Hole {\leaders\pattern\hfil} 1}
\line{The Pool of Tears {\leaders\pattern\hfil} 9}
\line{A Caucus-Race and a Long Tale {\cleaders\pattern
\hfil} 19}
\line{Pig and Pepper {\xleaders\pattern\hfil} 27}

```

```

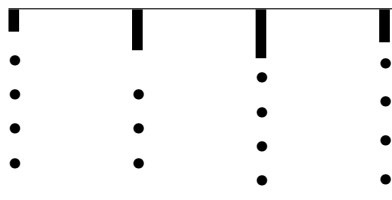
Down the Rabbit-Hole 1
The Pool of Tears 9
A Caucus-Race and a Long Tale 19
Pig and Pepper 27

```

```

\def\bulletfill{\vbox to 3ex{\vfil\hbox{\bullet}\vfil}}%
\def\mybox{\vbox to 1in}
\def\myrule{\hrule width 4pt}\hsize=2in
\hrule \line{%
\mybox{\myrule depth 8pt \leaders\bulletfill\vfill}
\hfil
\mybox{\myrule depth 15pt \leaders\bulletfill\vfill}
\hfil
\mybox{\myrule depth 18pt \cleaders\bulletfill\vfill}
\hfil
\mybox{\myrule depth 12pt \xleaders\bulletfill\vfill}%
}\hrule

```



```

\dotfill
\hrulefill
\dotfill \hrulefill

```

```
\hbox to 3in{开始 {\dotfill} 结束}
\hbox to 3in{瑞典语 {\hrulefill} 芬兰语}
```

```
.....
```

---

```
\leftarrowfill
\rightarrowfill
```

```
\hbox to 3in{\vrule \rightarrowfill \ 3 in
\leftarrowfill\vrule}
```

```
|—————→ 3 in ←————|
```

■

```
\+ <text> & <text> & ... \cr
\tabalign
\+ \tabalign \+ TEX “
‘&’ ‘&’
```

```

\cleartabs % Nullify any previous \settabs.
\+ {\bf if }$a[i] < a[i+1]$ &{\bf then}&\cr
\+&&$a[i] := a[i+1]$;\cr
\+&&{\it found }$:= $ {\bf true};\cr
\+&{\bf else}\cr
\+&&{\it found }$:= $ {\bf false};\cr
\+&{\bf end if};\cr

```

```

if $a[i] < a[i + 1]$ then
 $a[i] := a[i + 1];$
 found := true;
else
 found := false;
end if;

```

```

\settabs <number> \columns
\settabs \+ <sample line> \cr
 TEX <number> TEX \hsize \hsize

```

```

{\hsize = 3in \settabs 3 \columns
\+1& 一 & 第一 \cr
\+2& 二 & 第二 \cr
\+3& 三 & 第三 \cr}

```

```

1
2
3
|-----| 3 in
'&' TEX \cr
\settabs \settabs \settabs

```

```
% The first line establishes the template.
\settabs \+1\quad & three\quad & seventh\cr
\+1& 一 & 第一 \cr
\+2& 二 & 第二 \cr
\+3& 三 & 第三 \cr
```

```
1
2
3
```

```
\cleartabs
```

```
\cr\endline\cr\cr 'endline'
```

■

```
\halign { <preamble> \cr <row> \cr ... <row> \cr }
\halign to <dimen> { <preamble> \cr <row> \cr ... <row> \cr }
\halign spread <dimen> { <preamble> \cr <row> \cr ... <row> \cr }
```

TEX

TEX “

‘&’ \cr ‘&’ \cr ‘#’ TEX \settabs ‘#’

TEX hbox

to TEX <dimen> spread TEX <dimen> \hbox ‘hbox’

\tabskip ‘tabskip’ to

```
\tabskip = 1em \halign{%
```

```
\hfil\it#\hfil&\hfil#\hfil&#\hfil\$\#\cr
```

```
& 华盛顿 & 美元 &1.00\cr
```

```
& 巴黎 & 法郎 &0.174\cr
```

```
& 耶路撒冷 & 新谢克尔 &0.507\cr
```

```
& 东京 & 日元 &0.0829\cr}
```

\$1.00  
 \$0.174  
 \$0.507  
 \$0.0829

```
\valign { <preamble>\cr <column>\cr ... <column>\cr }
\valign to <dimen>{ <preamble>\cr <column>\cr ... <column>\cr }
\valign spread <dimen>{ <preamble>\cr <column>\cr ... <column>\cr }
TEX
```

T<sub>E</sub>X a) 在概念上有一些困难，并且 (b) 不经常使，因此，我们建议在你试图使用 `\valign` 命令之前首先学习一下常见的阵列（第 ‘[1](#)’ 页）以及 `\halign` 命令（参见上面的描述）。

阵列由导言和随后的需要排列的文字构成。导言描述随后列的版式，由一系列行的模板构成，列之间由 ‘&’ 分隔，用 `\cr` 作为结束标志。每一列由一系列行的条目构成，也由 ‘&’ 分隔，用 `\cr` 作为结束标志。在模板中，‘#’ T<sub>E</sub>X

```
TEX vbox TEX TEX \parindent ‘parindent’ \leftskip
\rightskip ‘leftskip’
\hsize ‘hsize’ \hsize
\strut
to TEX <dimen> spread TEX <dimen> \vbox ‘vbox’
```

```
{\hsize=1in \parindent=0pt
\valign{#\strut&#\strut&#\strut&#\strut\cr
 bernaise&curry&hoisin&hollandaise\cr
 ketchup&marinara&mayonnaise&mustard\cr
 rarebit&tartar\cr}}
```

|             |            |         |
|-------------|------------|---------|
| bernaise    | ketchup    | rarebit |
| curry       | marinara   | tartar  |
| hoisin      | mayonnaise |         |
| hollandaise | mustard    |         |

```

%
% (这个例子显示了支架存在的必要性)。
{\hsize=1in \parindent=0pt
\valign{#&#&#&#&#\cr
 bernaise&curry&hoisin&hollandaise\cr
 ketchup&marinara&mayonnaise&mustard\cr
 rarebit&tartar\cr}}

 bernaise ketchup rarebit
 curry marinara tartar
 hoisin mayonnaise
 hollandaise mustard

\ialign
 \tabskip \everycr \halign

\cr
 \everycr ‘everycr’ TEX \cr

\endline
 \cr \cr

\crrcr
 \cr \noalign TEX \cr \cr \crrcr \cr

\omit
 TEX \omit ‘#’

\tabskip = 2em\halign{%
 \hfil\it#\hfil&\hfil#\hfil&#\hfil\$\#\cr
 & 华盛顿 & 美元 &1.00\cr
 \omit \dotfill \dotfill& 巴黎 & 法郎 &0.174\cr
 & 耶路撒冷 & 新谢克尔 &0.507\cr
 & 东京 & 日元 &0.0829\cr}

```



\$1.00  
\$0.174  
\$0.507  
\$0.0829

```
{\hsize=1.2in \parindent=0pt
\valign{(\#)\strut&(\#)\strut&(\#)\strut&(\#)\strut\cr
 bernaise&curry&hoisin&hollandaise\cr
 ketchup&\omit\strut{\bf MARINARA!}&mayonnaise&mustard\cr
 rarebit&tartar\cr}}
```

|               |                  |           |
|---------------|------------------|-----------|
| (bernaise)    | (ketchup)        | (rarebit) |
| (curry)       | <b>MARINARA!</b> | (tartar)  |
| (hoisin)      | (mayonnaise)     |           |
| (hollandaise) | (mustard)        |           |

`\span`

- $\text{\TeX}$  `\span`  $\text{\TeX}$
- `\span` ‘&’, 放在两列或两行条目之前，会导致这些列或行合并到一起。对于水平阵列，合并的列的宽度是列成员宽度之和。对于竖直阵列，合并的行的高度是行成员高度之和。合并的行或合并的列的模板形成一个编组，因此，在 `\span` 前的字体设置命令影响下一个 ‘&’ 前的一切。

`\span` `\multispan`

`\multispan`  $\langle number \rangle$

$\text{\TeX}$   $\langle number \rangle$   $\langle number \rangle$  `\span` `\omit`

```
\tabskip = 13pt\halign{%
 \hfil\it#\hfil&\hfil#\hfil&#\hfil\$\#\cr
 & 华盛顿 & 美元 &1.00\cr
 & 巴黎 & 法朗 &0.174\cr
 & 耶路撒冷 &
 \multispan 2 \hfil\it(无信息)\hfil \cr
 & 东京 & 日元 &0.0829\cr}
```

\$1.00  
\$0.174  
(无信息)  
\$0.0829

```
{\hsize=1.2in \parindent=0pt
\valign{(\#)\strut&(\#)\strut&(\#)\strut&(\#)\strut\cr
 bernaise&curry&hoisin&hollandaise\cr
 \multispan 3$\left\{\left\{\rm ketchup\right\}\atop{\rm marinara}\right\}$$\&mustard\cr
 rarebit&tartar\cr}}
```

|               |             |              |
|---------------|-------------|--------------|
| (bernaise)    |             | (rarebit)    |
| (curry)       | { ketchup } | (tartar)     |
| (hoisin)      |             | { marinara } |
| (hollandaise) | (mustard)   |              |

```
\noalign { \langle vertical mode material \rangle }
\noalign { \langle horizontal mode material \rangle }
 \langle vertical mode material \rangle \langle horizontal mode material \rangle
\noalign \openup ‘openup’
```

```
\halign{%
 \hfil\it#\hfil\tabskip=2em&\hfil#\hfil&##
 \hfil\$\#\tabskip=0em\cr
 % \tabskip
 & 华盛顿 & 美元 &1.00\cr
 & 巴黎 & 法郎 &0.174\cr
 \noalign{\smallskip\hrule\smallskip}
 & 耶路撒冷 & 新谢克尔 &0.507\cr
 & 东京 & 日元 &0.0829\cr}
```

|       |          |
|-------|----------|
|       | \$1.00   |
|       | \$0.174  |
| <hr/> |          |
|       | \$0.507  |
|       | \$0.0829 |

```
{\hsize=1in \parindent=0pt
\valign{#\strut&\strut&\strut&\strut\cr
 \noalign{\vrule width 2pt\quad}
 bernaise&curry&hoisin&hollandaise\cr
 \noalign{\vrule width 2pt\quad}
 ketchup&marinara&mayonnaise&mustard\cr
 \noalign{\vrule width 2pt\quad}
 rarebit&tartar\cr
 \noalign{\vrule width 2pt\quad}}}
```

|  |             |  |            |  |         |  |
|--|-------------|--|------------|--|---------|--|
|  | bernaise    |  | ketchup    |  | rarebit |  |
|  | curry       |  | marinara   |  | tartar  |  |
|  | hoisin      |  | mayonnaise |  |         |  |
|  | hollandaise |  | mustard    |  |         |  |

`\tabskip` [*⟨glue⟩* parameter]

`TeXTeX \tabskip \tabskip---` 这一改变将影响所有紧接其后的 `&` 的粘连以及最后一行或列之后的粘连。

```

\halign to 3.5in{%
 \hfil\it#\tabskip = 2em plus 8pt
 \hfil&\hfil#\hfil&\tabskip = 1em
 &\hfil\$\#\tabskip = 0em\cr
 & 华盛顿 & 美元 &1.00\cr
 & 巴黎 & 法郎 &0.174\cr
 & 耶路撒冷 & 新谢克尔 &0.507\cr
 & 东京 & 日元 &0.0829\cr}

$1.00
$0.174
$0.507
$0.0829

```

```

{\hsize = 1in \parindent=0pt \tabskip=5pt
\valign{#\strut&\strut\tabskip = 3pt
 &\strut&\strut\cr
 bernaise&curry&hoisin&hollandaise\cr
 ketchup&marinara&mayonnaise&mustard\cr
 rarebit&tartar\cr}}

```

|             |            |         |
|-------------|------------|---------|
| bernaise    | ketchup    | rarebit |
| curry       | marinara   | tartar  |
| hoisin      | mayonnaise |         |
| hollandaise | mustard    |         |

```

\hidewidth
TEX \hidewidth \hidewidth

```

```
\tabskip = 25pt\halign{%
 \hfil\it#\hfil&\hfil#\hfil&#\hfil\$\#\cr
 &\hidewidth & 美元 &1.00\cr
 & 巴黎 & 法郎 &0.174\cr
 & 耶路撒冷 & 新谢克尔 &0.507\cr
 & 东京 & 日元 &0.0829\cr}
```

\$1.00  
\$0.174  
\$0.507  
\$0.0829

```
\everycr [<token list> parameter]
```

```
TeX\cr <token list>\everycr \cr \everycr TeX
\everycr \noalign\everycr \cr \noalign TeX TeX \cr \cr
\everycr
```

```
\everycr={\noalign{\smallskip\hrule\smallskip}}
\halign{#\tabskip = 11pt&\hfil#\hfil&\hfil#\hfil
 \tabskip = 0pt\cr
 1& 一 & 第一 \cr
 2& 二 & 第二 \cr
 3& 三 & 第三 \cr}
```

---

1

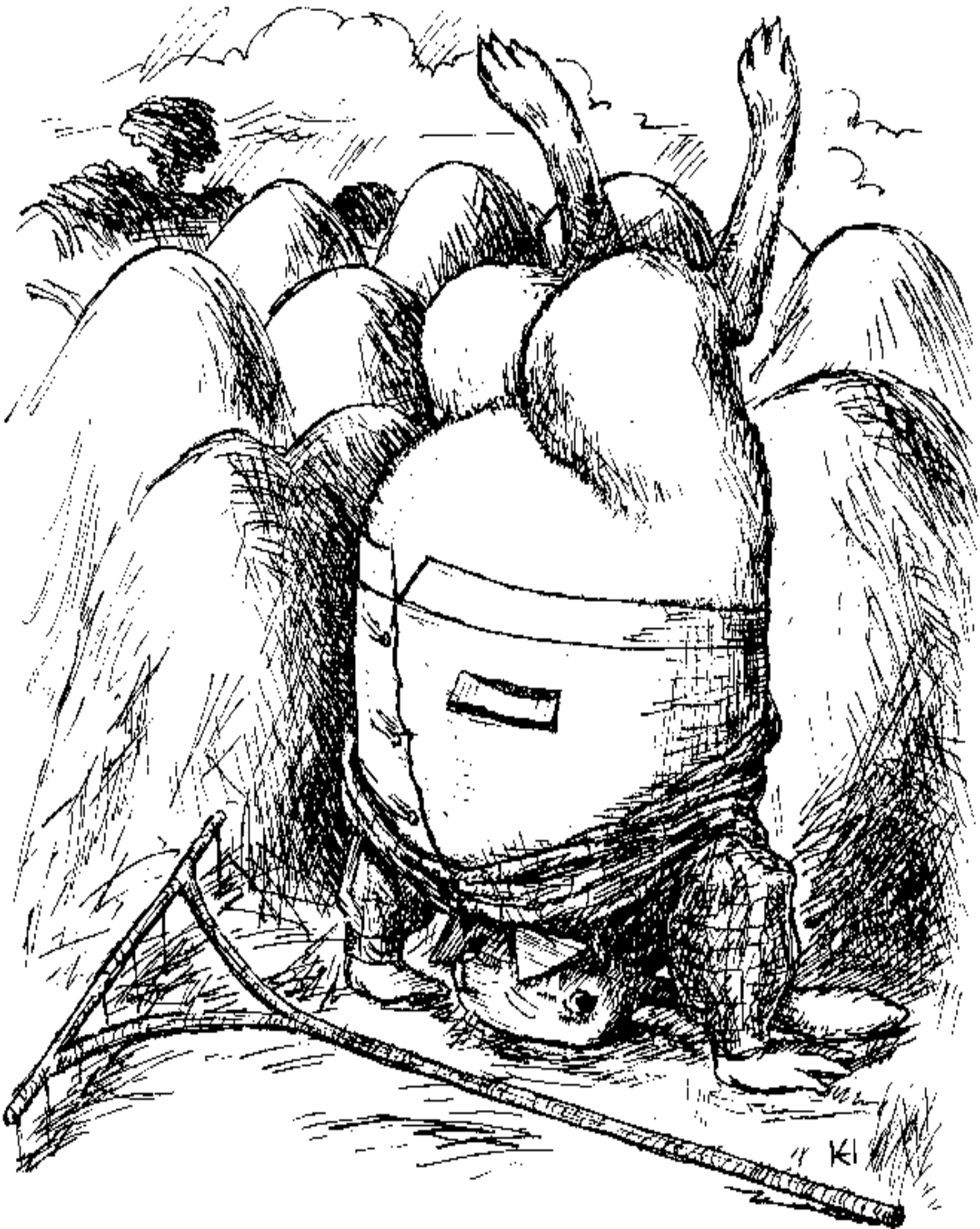
---

2

---

3

---







`'cmddesc'`



|   |               |                          |           |                      |             |                        |
|---|---------------|--------------------------|-----------|----------------------|-------------|------------------------|
| ☆ | $\alpha$      | <code>\alpha</code>      | $\mu$     | <code>\mu</code>     | $\sigma$    | <code>\sigma</code>    |
|   | $\beta$       | <code>\beta</code>       | $\nu$     | <code>\nu</code>     | $\varsigma$ | <code>\varsigma</code> |
|   | $\chi$        | <code>\chi</code>        | $\omega$  | <code>\omega</code>  | $\Sigma$    | <code>\Sigma</code>    |
|   | $\delta$      | <code>\delta</code>      | $\Omega$  | <code>\Omega</code>  | $\tau$      | <code>\tau</code>      |
|   | $\Delta$      | <code>\Delta</code>      | $\phi$    | <code>\phi</code>    | $\theta$    | <code>\theta</code>    |
|   | $\epsilon$    | <code>\epsilon</code>    | $\varphi$ | <code>\varphi</code> | $\vartheta$ | <code>\vartheta</code> |
|   | $\varepsilon$ | <code>\varepsilon</code> | $\Phi$    | <code>\Phi</code>    | $\Theta$    | <code>\Theta</code>    |
|   | $\eta$        | <code>\eta</code>        | $\pi$     | <code>\pi</code>     | $\upsilon$  | <code>\upsilon</code>  |
|   | $\gamma$      | <code>\gamma</code>      | $\varpi$  | <code>\varpi</code>  | $\Upsilon$  | <code>\Upsilon</code>  |
|   | $\Gamma$      | <code>\Gamma</code>      | $\Pi$     | <code>\Pi</code>     | $\xi$       | <code>\xi</code>       |
|   | $\iota$       | <code>\iota</code>       | $\psi$    | <code>\psi</code>    | $\Xi$       | <code>\Xi</code>       |
|   | $\kappa$      | <code>\kappa</code>      | $\Psi$    | <code>\Psi</code>    | $\zeta$     | <code>\zeta</code>     |
|   | $\lambda$     | <code>\lambda</code>     | $\rho$    | <code>\rho</code>    |             |                        |
|   | $\Lambda$     | <code>\Lambda</code>     | $\varrho$ | <code>\varrho</code> |             |                        |

(\$) T<sub>E</sub>X `{\rm o}` omicron ‘o’, `{\rm B}` beta (‘B’).



- `\upsilon` ( $\upsilon$ ), `\rm v` ( $\mathrm{v}$ ), `\nu` ( $\nu$ ).
- `\varsigma` ( $\varsigma$ ) `\zeta` ( $\zeta$ ).

`(\mit)`  
`\TeX`

`\rho$ $ \theta$ $ f(\theta)`  
`-\{\mit \Gamma\}_{\theta} < f(\rho)-\{\mit \Gamma\}_{\rho}$.`

$$\rho \quad \theta \quad f(\theta) - \Gamma_\theta < f(\rho) - \Gamma_\rho.$$

■

|                                      |                                |                                          |
|--------------------------------------|--------------------------------|------------------------------------------|
| ☆ $\infty$ <code>\infty</code>       | $\exists$ <code>\exists</code> | $\partial$ <code>\partial</code>         |
| $\Re$ <code>\Re</code>               | $\forall$ <code>\forall</code> | $\sqrt{\phantom{x}}$ <code>\sqrt</code>  |
| $\Im$ <code>\Im</code>               | $\hbar$ <code>\hbar</code>     | $\wp$ <code>\wp</code>                   |
| $\angle$ <code>\angle</code>         | $\ell$ <code>\ell</code>       | $\flat$ <code>\flat</code>               |
| $\triangle$ <code>\triangle</code>   | $\aleph$ <code>\aleph</code>   | $\sharp$ <code>\sharp</code>             |
| $\backslash$ <code>\backslash</code> | $\imath$ <code>\imath</code>   | $\natural$ <code>\natural</code>         |
| $\mid$ <code>\mid</code>             | $j$ <code>\jmath</code>        | $\clubsuit$ <code>\clubsuit</code>       |
| $\parallel$ <code>\parallel</code>   | $\nabla$ <code>\nabla</code>   | $\diamondsuit$ <code>\diamondsuit</code> |
| $\Vdash$ <code>\Vdash</code>         | $\neg$ <code>\neg</code>       | $\heartsuit$ <code>\heartsuit</code>     |
| $\emptyset$ <code>\emptyset</code>   | $\nmid$ <code>\nmid</code>     | $\spadesuit$ <code>\spadesuit</code>     |
| $\perp$ <code>\bot</code>            | $'$ <code>'</code> (上标点)       |                                          |
| $\top$ <code>\top</code>             | $'$ <code>\prime</code>        |                                          |

`(\$)`  
`'i' 'j' \imath \jmath`  
`(') \prime (\prime)`  
`\| \Vdash \neg \nmid \mid ' |`  
`\backslash, \mid, \Vdash . \bigm (第 \bigm 页)`

The Knave of  $\heartsuit$ s, he stole some tarts.

The Knave of  $\hearts$ s, he stole some tarts.

$\hat{i} < \hat{j}$   $i' \leq j'$ .

$\hat{i} < \hat{j}$   $i' \leq j'$ .

$\frac{x-a}{x+a} \backslash \frac{y-b}{y+b}$

$$\frac{x-a}{x+a} \backslash \frac{y-b}{y+b}$$

■

|                                  |                                  |                                                  |
|----------------------------------|----------------------------------|--------------------------------------------------|
| ☆ $\vee$ <code>\vee</code>       | $\cdot$ <code>\cdot</code>       | $\triangleleft$ <code>\triangleleft</code>       |
| $\wedge$ <code>\wedge</code>     | $\diamond$ <code>\diamond</code> | $\triangleright$ <code>\triangleright</code>     |
| $\amalg$ <code>\amalg</code>     | $\bullet$ <code>\bullet</code>   | $\bigtriangledown$ <code>\bigtriangledown</code> |
| $\cap$ <code>\cap</code>         | $\circ$ <code>\circ</code>       | $\triangleup$ <code>\triangleup</code>           |
| $\cup$ <code>\cup</code>         | $\bigcirc$ <code>\bigcirc</code> | $*$ <code>\ast</code>                            |
| $\uplus$ <code>\uplus</code>     | $\odot$ <code>\odot</code>       | $\star$ <code>\star</code>                       |
| $\sqcap$ <code>\sqcap</code>     | $\ominus$ <code>\ominus</code>   | $\times$ <code>\times</code>                     |
| $\sqcup$ <code>\sqcup</code>     | $\oplus$ <code>\oplus</code>     | $\div$ <code>\div</code>                         |
| $\dagger$ <code>\dagger</code>   | $\oslash$ <code>\oslash</code>   | $\setminus$ <code>\setminus</code>               |
| $\ddagger$ <code>\ddagger</code> | $\otimes$ <code>\otimes</code>   | $\wr$ <code>\wr</code>                           |
| $\wedge$ <code>\land</code>      | $\pm$ <code>\pm</code>           |                                                  |
| $\vee$ <code>\lor</code>         | $\mp$ <code>\mp</code>           |                                                  |

$\text{\TeX}$  .  $\text{\TeX}$   $\text{\TeX}$  ---不过仅在它出现在公式的最外层时，而不是在一个组中。

除了这些命令以外， $\text{\TeX}$  ‘+’ and ‘-’ ‘/’

$z = x \div y \quad \text{\hbox{当且仅当}} \quad z \times y = x \quad y \neq 0$

$$z = x \div y \quad z \times y = x \quad y \neq 0$$

$\backslash *$

$\backslash *$  ( $\times$ )，它是一个二元运算符。该符号在文内公式中功能与自定连字符类似。这就是说，公式需要在该处断行时， $\text{\TeX}$   $\backslash \times$   $\text{\TeX}$   $\backslash *$

Let  $c = a * b$ . In the case that  $c=0$  or  $c=1$ , let  $\Delta$  be  $(\text{the smallest } q) * (\text{the largest } q)$  in the set of approximate  $\tau$ -values.

Let  $c = ab$ . In the case that  $c = 0$  or  $c = 1$ , let  $\Delta$  be (the smallest  $q) \times$  (the largest  $q)$  in the set of approximate  $\tau$ -values.

■

|             |           |               |
|-------------|-----------|---------------|
| ☆ $\asymp$  | $\gg$     | $\bowtie$     |
| $\cong$     | $\ll$     | $\propto$     |
| $\dashv$    | $\models$ | $\approx$     |
| $\vdash$    | $\neq$    | $\sim$        |
| $\perp$     | $\neq$    | $\simeq$      |
| $\mid$      | $\notin$  | $\frown$      |
| $\parallel$ | $\in$     | $\smile$      |
| $\doteq$    | $\ni$     | $\subset$     |
| $\equiv$    | $\owns$   | $\subseteq$   |
| $\geq$      | $\prec$   | $\supset$     |
| $\geq$      | $\preceq$ | $\supseteq$   |
| $\leq$      | $\succ$   | $\sqsubseteq$ |
| $\leq$      | $\succeq$ | $\sqsupseteq$ |

$\text{\TeX}$   $\text{\TeX}$   $\text{\TeX}$  ---不过仅在它出现在公式的最外层时，而不是在一个组中。

除了这里列出的命令以外， $\text{\TeX}$  ‘=’ “箭头” (第 `arrows` 页)

- ‘ $\geq$ ’ (`\geq` `\geq`).
- ‘ $\leq$ ’ (`\leq` `\leq`).
- ‘ $\neq$ ’ (`\neq`, `\neq`, `\not=`).
- ‘ $\ni$ ’ (`\ni` `\owns`).

$\text{\not}$ ,

|              |               |                   |
|--------------|---------------|-------------------|
| $\not\asymp$ | $\not\leq$    | $\not\simeq$      |
| $\not\cong$  | $\not\prec$   | $\not\subset$     |
| $\not\equiv$ | $\not\preceq$ | $\not\subseteq$   |
| $\not=$      | $\not\succ$   | $\not\supset$     |
| $\not\geq$   | $\not\succeq$ | $\not\supseteq$   |
| $\not\geq$   | $\not\approx$ | $\not\sqsubseteq$ |
| $\not\leq$   | $\not\sim$    | $\not\sqsupseteq$ |

$\$AB \perp AC$ , 且

=

$\$ \triangle ABF \not\sim \triangle ACF$ .

$AB \perp AC$ , 且  $\triangle ABF \not\sim \triangle ACF$ .

■

|     |         |   |         |   |         |
|-----|---------|---|---------|---|---------|
| ☆ { | \lbrace | [ | \lbrack | [ | \lceil  |
| {   | \{      | ] | \rbrack | ] | \rceil  |
| }   | \rbrace | < | \langle | ] | \lfloor |
| }   | \}      | > | \rangle | ] | \rfloor |

“”, 右定界符又称为 “”。开符号和闭符号是  $\text{T}_\text{E}\text{X}$  数学公式中的两种字符类。 $\text{T}_\text{E}\text{X}$  在不同类的数学符号之间留下不同大小的间隔。你也许认为在开符号和闭符号旁边的间隔是对称的, 但实际上并非如此。

有些左定界符和右定界符可以用不止一个命令排印:

- `\{'` (`\lbrace` 和 `\{`)
- `\}'` (`\rbrace` 和 `\}`)
- `\['` (`\lbrack` 和 `\lceil`)
- `\]'` (`\rbrack` 和 `\rfloor`)

左右方括号 (两种形式皆可) 在数学模式之外也可以使用。

除这些命令之外,  $\text{T}_\text{E}\text{X}$  ‘(’ ‘)’

`\left \right` ‘left’  $\text{T}_\text{E}\text{X}$  `\bigx \big` ‘big’

$$\${f\{x \mid x > 0\}, \}$$

$$\{x \mid x > 0\}$$

■

|     |                                  |    |                                 |
|-----|----------------------------------|----|---------------------------------|
| ☆ ← | <code>\leftarrow</code>          | ←  | <code>\leftharpoondown</code>   |
| ←   | <code>\gets</code>               | →  | <code>\rightharpoondown</code>  |
| ⇐   | <code>\Leftarrow</code>          | ⇐  | <code>\leftharpoonup</code>     |
| →   | <code>\rightarrow</code>         | →  | <code>\rightharpoonup</code>    |
| →   | <code>\to</code>                 | ⇌  | <code>\rightleftharpoons</code> |
| ⇒   | <code>\Rightarrow</code>         | ↦  | <code>\mapsto</code>            |
| ↔   | <code>\leftrightarrow</code>     | ↦→ | <code>\longmapsto</code>        |
| ⇔   | <code>\Leftrightarrow</code>     | ↓  | <code>\downarrow</code>         |
| ←   | <code>\longleftarrow</code>      | ↓  | <code>\Downarrow</code>         |
| ⇐   | <code>\Longleftarrow</code>      | ↑  | <code>\uparrow</code>           |
| →   | <code>\longrightarrow</code>     | ↑  | <code>\Uparrow</code>           |
| ⇒   | <code>\Longrightarrow</code>     | ↑  | <code>\updownarrow</code>       |
| ↔   | <code>\longleftrightarrow</code> | ↕  | <code>\Updownarrow</code>       |
| ⇔   | <code>\Longleftrightarrow</code> | ↗  | <code>\nearrow</code>           |
| ⇔   | <code>\iff</code>                | ↘  | <code>\searrow</code>           |
| ↵   | <code>\hookleftarrow</code>      | ↖  | <code>\nwarrow</code>           |
| ↷   | <code>\hookrightarrow</code>     | ↗  | <code>\swarrow</code>           |

```

‘relations’ \big ‘big’
\iff \Longleftrightarrow
\buildrel ‘buildrel’

```

$$f(x) \mapsto f(y) \iff x \mapsto y$$

$$f(x) \mapsto f(y) \iff x \mapsto y$$

■

☆

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

TeX `\det\gcd\inf\lim\liminf\limsup\max\min\Pr\sup` TeX

$$\cos^2 x + \sin^2 x = 1 \quad \max_{a \in A} g(a) = 1$$

$$\cos^2 x + \sin^2 x = 1 \qquad \max_{a \in A} g(a) = 1$$

`\bmod`

$$x = (y+1) \bmod 2$$

$$x = (y + 1) \bmod 2$$

`\pmod`

$$x \equiv y+1 \pmod 2$$

$$x \equiv y + 1 \pmod 2$$



|   |              |              |                         |             |             |                        |         |         |                        |
|---|--------------|--------------|-------------------------|-------------|-------------|------------------------|---------|---------|------------------------|
| ☆ | $\bigcap$    | $\bigcap$    | <code>\bigcap</code>    | $\bigsqcup$ | $\bigsqcup$ | <code>\bigsqcup</code> | $\int$  | $\int$  | <code>\smallint</code> |
|   | $\bigcup$    | $\bigcup$    | <code>\bigcup</code>    | $\biguplus$ | $\biguplus$ | <code>\biguplus</code> | $\int$  | $\int$  | <code>\int</code>      |
|   | $\bigodot$   | $\bigodot$   | <code>\bigodot</code>   | $\bigvee$   | $\bigvee$   | <code>\bigvee</code>   | $\oint$ | $\oint$ | <code>\oint</code>     |
|   | $\bigoplus$  | $\bigoplus$  | <code>\bigoplus</code>  | $\bigwedge$ | $\bigwedge$ | <code>\bigwedge</code> | $\prod$ | $\prod$ | <code>\prod</code>     |
|   | $\bigotimes$ | $\bigotimes$ | <code>\bigotimes</code> | $\coprod$   | $\coprod$   | <code>\coprod</code>   | $\sum$  | $\sum$  | <code>\sum</code>      |

`\TeX` `\TeX` `\TeX`

‘`\big`’ `\cap` ( $\cap$ ), 见第 [‘binops’](#) 页) 不同, 因为它们通常出现公式的开头。TeX 给巨算符留下的间隔与二元运算符的不同。

不要混淆 `\sum` ( $\sum$ ) 和 `\Sigma` ( $\Sigma$ ), 或者 `\prod` ( $\prod$ ) 和 `\Pi` ( $\Pi$ )。 `\Sigma` 和 `\Pi` 排印大写希腊字母, 它们尺寸更小, 外观也不同。

巨算符可以带有极限。下极限用下标指定, 而上极限用上标指定。

`$$\bigcap_{k=1}^r (a_k \cup b_k)$$`

$$\bigcap_{k=1}^r (a_k \cup b_k)$$

`$$\{\int_0^\pi \sin^2 ax \, dx\} = \{\pi \over 2\}$$`

$$\int_0^\pi \sin^2 ax \, dx = \frac{\pi}{2}$$

`\limits`

`\TeX` `\TeX`



```
\limits\nolimits \displaylimits
```

Suppose that  $\bigcap_{i=1}^N q_i$  contains at least two elements.

Suppose that  $\bigcap_{i=1}^N q_i$  contains at least two elements.

```
\nolimits
TeX TeX
\int \oint \int\limits TeX plain TeX \int \oint \nolimits
\limits\nolimits \displaylimits
```

```
$$\bigcap_{i=1}^N q_i$$
```

$$\bigcap_{i=1}^N q_i$$

```
\displaylimits
TeX
\limits\nolimits \displaylimits
```

```
$$a(\lambda) = \{1 \over {2\pi}\} \int\limits_{-\infty}^{+\infty} f(x)e^{-i\lambda x}\,dx$$
```

$$a(\lambda) = \frac{1}{2\pi} \int\limits_{-\infty}^{+\infty} f(x)e^{-i\lambda x} dx$$



```
\cdotp
\ldotp
TeX \cdot ‘\cdot’ TeX
```

`$x \cdotp y \quad x \ldotp y \quad x \cdot y$`

$x \cdot y \quad x.y \quad x \cdot y$

`\colon`

`\colon (:)` 的区别在于, ``:'` 是一个运算符, 因此 `TEX` 在其左侧留下额外间隔, 然而在 `\colon` 左侧却不留额外间隔。

`$f \colon t \quad f : t$`

$f:t \quad f:t$

☆ `_ <argument>`                      `^ <argument>`  
`\sb <argument>`                      `\sp <argument>`

`<argument> <argument> \sb \sp ‘_’ or ‘^’`

`TEX a_i_j a^i_j a^i_j`

`\scriptstyle \scriptscriptstyle ‘scriptscriptstyle’`

`‘namedfns’ ‘namedfns’ \lim \det`

`$x_3 \quad t_{\max} \quad a_{i_k} \quad \sum_{i=1}^n q_i \quad x^3 \quad e^{t \cos \theta} \quad r^{x^2} \quad \int_0^\infty f(x) dx`

`\quad x^3 \quad e^{\cos \theta} \quad r^{x^2} \quad \int_0^\infty f(x) dx`

`\int_0^\infty f(x) dx`

`$$$ \lim_{x \rightarrow 0} f(x) \quad \det^{z \in A} \quad \sin^2 t $$$`

$x_3 \quad t_{\max} \quad a_{i_k} \quad \sum_{i=1}^n q_i \quad x^3 \quad e^{t \cos \theta} \quad r^{x^2} \quad \int_0^\infty f(x) dx$

$\lim_{x \rightarrow 0} f(x) \quad \det^{z \in A} \quad \sin^2 t$



```
\textstyle
\scriptstyle
\scriptscriptstyle
\displaystyle
\TeX \it \TeX
```

```
$t+{\scriptstyle t} + {\scriptscriptstyle t}$
```

```
 $t + t + t$
```

```
\mathchoice {⟨math_1⟩} {⟨math_2⟩} {⟨math_3⟩} {⟨math_4⟩}
\TeX ⟨math_1⟩⟨math_2⟩⟨math_3⟩ ⟨math_4⟩ \TeX \mathchoice ⟨math_1⟩
⟨math_2⟩ ⟨math_3⟩ ⟨math_4⟩
```

```
\def\mc{{\mathchoice{D}{T}{S}{SS}}}
The strange formula $\mc_{\mc\mc}$ illustrates a
mathchoice.
```

The strange formula  $T_{SS}$  illustrates a mathchoice.

```
\mathpalette ⟨argument_1⟩ ⟨argument_2⟩
1 \build \mathpalette \mathpalette\build⟨argument⟩
\build \TeX ⟨argument⟩ \mathpalette #1 #2 ⟨argument⟩ \build
\build ‘#1’ The \TeXbook 360 \mathpalette The \TeXbook 151
```

---

<sup>1</sup> `\def\mathpalette#1#2{\mathchoice{#1\displaystyle{#2}}{#1\textstyle{#2}}{#1\scriptstyle{#2}}{#1\scriptscriptstyle{#2}}}`





```

\overwithdelims <delim_1> <delim_2>
\atopwithdelims <delim_1> <delim_2>
\abovewithdelims <delim_1> <delim_2> <dimen>
 <delim_1> <delim_2> \over\atop \above \abovewithdelims <dimen>

```

```

$$\{m \overwithdelims () n\}\qquad
\{m \atopwithdelims || n\}\qquad
\{m \abovewithdelims \{\}\ 2pt n\}$$

```

$$\left(\frac{m}{n}\right) \quad \left|m\right| \quad \left\{\frac{m}{n}\right\}$$

```

\cases
 ‘&’ TEX \cr

```

```

$$g(x,y) = \cases{f(x,y),&if $x<y$\cr
 f(y,x),&if $x>y$\cr
 0,&otherwise.\cr}$$

```

$$g(x,y) = \begin{cases} f(x,y), & \text{if } x < y \\ f(y,x), & \text{if } x > y \\ 0, & \text{otherwise.} \end{cases}$$

```

\underbrace <argument>
\overbrace <argument>
\underline <argument>
\overline <argument>
\overleftarrow <argument>
\overrightarrow <argument>
 <argument> TEX TEX <argument> \mathstrut ‘mathstrut’

```

```
$$\displaylines{
\underbrace{x \circ y}\qquad \overbrace{x \circ y}\qquad
\underline{x \circ y}\qquad \overline{x \circ y}\qquad
\overleftarrow{x \circ y}\qquad
\overrightarrow{x \circ y}\cr
{\overline r + \overline t}\qquad
{\overline {r \mathstrut} + \overline {t \mathstrut}}}\cr
}$$
```

$$\underbrace{x \circ y} \qquad \overbrace{x \circ y} \qquad \underline{x \circ y} \qquad \overline{x \circ y} \qquad \overleftarrow{x \circ y} \qquad \overrightarrow{x \circ y}$$
$$\overline{r} + \overline{t} \qquad \overline{r} + \overline{t}$$

```
\buildrel <formula> \over <relation>
<formula> <relation> TEX “
```

```

\buildrel \rm def \over \equiv$

\def
\equiv
```



```
☆ \ldots
\cdots
\ldots \cdots ‘vcenter’ \vcenter
```

```
$t_1 + t_2 + \cdots + t_n \qquad x_1,x_2, \ldots, x_r$
```

$$t_1 + t_2 + \cdots + t_n \qquad x_1,x_2,\ldots,x_r$$

```
☆ \vdots
```

```


$$\begin{aligned} f(\alpha_1) &= f(\beta_1) \\ &\vdots \\ f(\alpha_k) &= f(\beta_k) \end{aligned}$$


```

$$f(\alpha_1) = f(\beta_1)$$

$$\vdots$$

$$f(\alpha_k) = f(\beta_k)$$

`\ddots`

```


$$\begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$


```

$$\begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

`\dots` ‘dots’

■

`\lgroup`

`\rgroup`

`\Big`

```


$$\dots \quad \bigg(\dots \bigg)$$


```

$$(\dots) \quad \left( \dots \right)$$





```

\matrix { \langle line \rangle \cr ... \langle line \rangle \cr }
\pmatrix { \langle line \rangle \cr ... \langle line \rangle \cr }
\bordermatrix { \langle line \rangle \cr ... \langle line \rangle \cr }

```

这三个命令每个都排印一个矩阵，输入矩阵时各行的元素之间用`&'分隔，而各行用\cr结尾。（这里使用与阵列一样的形式。）这些命令之间的区别如下：

- \matrix 排印一个四周空白不带定界符的矩阵。
- \pmatrix 排印一个两边带圆括号的矩阵。
- \bordermatrix 排印一个将第一行和第一列视为标号的矩阵。（第一行的第一个元素通常为空白。）矩阵的其他元素被圆括号包含。

通过增加竖直延伸，T<sub>E</sub>X 能够为 \pmatrix 和 \bordermatrix 制作足够大的圆括号。如果你需要用不同于圆括号的定界符包围矩阵，你应当将 \matrix 与 \left 和 \right（第`left'页）合起来使用。

```

$$\displaylines{
 \matrix{t_{11}&t_{12}&t_{13}\cr
 t_{21}&t_{22}&t_{23}\cr
 t_{31}&t_{32}&t_{33}\cr}\qquad
\left\{\matrix{t_{11}&t_{12}&t_{13}\cr
 t_{21}&t_{22}&t_{23}\cr
 t_{31}&t_{32}&t_{33}\cr}\right\}\cr
\pmatrix{t_{11}&t_{12}&t_{13}\cr
 t_{21}&t_{22}&t_{23}\cr
 t_{31}&t_{32}&t_{33}\cr}\qquad
\bordermatrix{&c_1&c_2&c_3\cr
 r_1&t_{11}&t_{12}&t_{13}\cr
 r_2&t_{21}&t_{22}&t_{23}\cr
 r_3&t_{31}&t_{32}&t_{33}\cr}\cr}

```

$$\begin{array}{ccc}
 t_{11} & t_{12} & t_{13} \\
 t_{21} & t_{22} & t_{23} \\
 t_{31} & t_{32} & t_{33}
 \end{array}
 \left\{
 \begin{array}{ccc}
 t_{11} & t_{12} & t_{13} \\
 t_{21} & t_{22} & t_{23} \\
 t_{31} & t_{32} & t_{33}
 \end{array}
 \right\}$$

$$\begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \quad \begin{matrix} c_1 & c_2 & c_3 \\ r_1 \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \\ r_2 \\ r_3 \end{matrix}$$

■

☆ `\sqrt`  $\langle argument \rangle$

此命令排印  $\langle argument \rangle$  的平方根。

`$$x = \{-b\pm\sqrt{b^2-4ac}\} \over 2a}$$`

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

☆ `\root`  $\langle argument_1 \rangle$  `\of`  $\langle argument_2 \rangle$

此命令排印  $\langle argument_2 \rangle$  的  $\langle argument_1 \rangle$  次根号。

`$$\root \alpha \of {r \cos \theta}$$`

$$\sqrt[\alpha]{r \cos \theta}$$

`\radical`  $\langle number \rangle$

此命令排印用  $\langle number \rangle$  刻画其特性的根数符号。它使用的定界码表示法与 `\delcode` 命令（第 ``delcode'` 页）的相同。

`\def\sqrt{\radical "270370} % as in plain TeX`

☆ `\eqno`

`\leqno`

这两个命令给陈列公式加上方程编号。`\eqno` 将编号放在右侧，而 `\leqno` 将编号放在左侧。这两个命令必须放在公式末尾。如果你有

个多行陈列公式，而你希望给不止一行编号， 你可以用 `\eqalignno` 或 `\leqalignno` 命令（第 ``eqalignno'` 页）。

这两个命令只能在陈列数学模式中使用。

```
$$e^{i\theta} = \cos \theta + i \sin \theta\eqno{(11)}$$
```

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (11)$$

```
$$\cos^2 \theta + \sin^2 \theta = 1\leqno{(12)}$$
```

$$(12) \qquad \cos^2 \theta + \sin^2 \theta = 1$$

```
\displaylines { \langle line \rangle \cr ... \langle line \rangle \cr }
```

此命令排印一个多行陈列公式，其中的各行独立地居中放置。你可以使用 `\noalign` 命令（第 ``noalign'` 页）改变多行陈列公式中两行的间隔。

如果你希望给多行陈列公式的某个或某些方程添加编号， 你应当使用 `\eqalignno` 或 `\leqalignno`。

```
$$\displaylines{(x+a)^2 = x^2+2ax+a^2\cr
(x+a)(x-a) = x^2-a^2\cr}$$
```

$$(x+a)^2 = x^2 + 2ax + a^2$$

$$(x+a)(x-a) = x^2 - a^2$$

```
\eqalign { \langle line \rangle \cr ... \langle line \rangle \cr }
\eqalignno { \langle line \rangle \cr ... \langle line \rangle \cr }
\leqalignno { \langle line \rangle \cr ... \langle line \rangle \cr }
```

这些命令排印一个多行陈列公式，其中某些行的对应部分竖直对齐。`\eqalignno` 和 `\leqalignno` 命令还允许你给某个或某些行添加方程编号。`\eqalignno` 将方程编号放在右侧，而 `\leqalignno` 将编号放在左侧。

陈列公式的每行用 `\cr` 结尾。各行需要对齐的各部分（多半是等号）前面加上 ``&'`。方程编号放在公式末尾，它的前面也要加上 ``&'`。你可以在单个陈列公式中多次使用这些命令以排印多组方程。在这种情形中，只有最右边或最左边的那组方程可以用 `\eqalignno` 或 `\leqalignno` 编号。

你可以使用 `\noalign` 命令（第 `'noalign'` 页）改变多行陈列公式中两行的间隔。

```


$$\left\{ \begin{array}{l} f_1(t) = 2t \\ f_2(t) = t^3 \\ f_3(t) = t^2 - 1 \end{array} \right\} \left\{ \begin{array}{l} g_1(t) = t \\ g_2(t) = 1 \end{array} \right.$$


```

$$\left\{ \begin{array}{l} f_1(t) = 2t \\ f_2(t) = t^3 \\ f_3(t) = t^2 - 1 \end{array} \right\} \left\{ \begin{array}{l} g_1(t) = t \\ g_2(t) = 1 \end{array} \right.$$

```


$$\sigma^2 = E(x - \mu)^2$$

$$= \frac{1}{n} \sum_{i=0}^n (x_i - \mu)^2$$

$$= E(x^2) - \mu^2$$


```

$$\begin{aligned} \sigma^2 &= E(x - \mu)^2 \\ &= \frac{1}{n} \sum_{i=0}^n (x_i - \mu)^2 \\ &= E(x^2) - \mu^2 \end{aligned} \tag{12}$$

```


$$\sigma^2 = E(x - \mu)^2$$

$$= E(x^2) - \mu^2$$


```

```

(6)
(7)

```

```


$$(x+a)^2 = x^2 + 2ax + a^2$$

$$(x+a)(x-a) = x^2 - a^2$$

% same effect as \displaylines but with an equation number

```

$$(x+a)^2 = x^2 + 2ax + a^2 \quad (19)$$

$$(x+a)(x-a) = x^2 - a^2$$

☆ `\cal` use calligraphic uppercase font  
`\mit` use math italic font  
`\oldstyle` use old style digit font

这些命令让  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  用指定的字体排版之后的文本。你只能在数学模式中使用它们。`\mit` 命令可用于排印斜体大写希腊字母。你还可以用 “” depth height (第 `'selfont'` 页) 中的那些命令改变数学模式中的字体。

```


$$\mathcal{XYZ} \quad \mathbb{AaBb} \quad \Gamma \Delta \Sigma \quad 0123456789$$


```

```


$$\mathcal{XYZ} \quad AaBb \Gamma \Delta \Sigma \quad 0123456789$$


```

```

\itfam family for italic type
\bffam family for boldface type
\slfam family for slanted type
\ttfam family for typewriter type

```

这些命令定义几种用于数学模式的字体族。它们主要用在 `\it`、`\bf`、`\sl` 和 `\tt` 命令的定义中，使这些命令能在数学模式中使用。

```

\fam [<number> parameter]

```

在数学模式时， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  通常用字符的数学码指定的字体族排版该字符。但是，如果  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  在数学模式中遇到第 7 类（变量）字符，它将由 `\fam` 的值给出的字体族排版该字符，只要 `\fam` 的值在 0 和 15 之间。如果 `\fam` 的值不在该范围内， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  就像通常情形那样使用字符的数学码指定的字体族。 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  在进入数学模式时设定 `\fam` 为 -1。在数学模式之外，`\fam` 无任何效果。

通过赋予 `\fam` 不同的值，你能让  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  用不同方式排版普通字符，比如变量。举个例子，设定了 `\fam` 为 `\ttfam`，你可以让  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  用打字机字体排版变量。Plain  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  在定义 `\tt` 宏时，除了其他设定之外，还设定 `\fam` 等于 `\ttfam`。

```

\def\bf{\fam\bffam\tenbf} % As in plain TeX.

```

```

\textfont <family> [<fontname> parameter]
\scriptfont <family> [<fontname> parameter]
\scriptscriptfont <family> [<fontname> parameter]

```

这三个参数分别选择  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  排版指定族的指定样式时所用的字体。这些选择在数学模式之外无任何效果。

```

\scriptfont2 = \sevensy % As in plain TeX.

```

“字体风格”（第 `\seltype` 页）。





```


$$\hbox to 1in{\downbracefill} \quad \hbox to 1in{\upbracefill}$$


```



```

\arrowvert
\Arrowvert
\lmoustache
\rmoustache
\bracevert

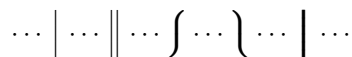
```

这些命令排印某些大定界符的一部分，把它们也用作定界符。它们取自 cmex10 数学字体中的字符。

```


$$\cdots \Big\arrowvert \cdots \Big\Arrowvert \cdots \Big\lmoustache \cdots \Big\rmoustache \cdots \Big\bracevert \cdots$$


```



```

\skew <number> <argument_1> <argument_2>

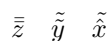
```

此命令将重音  $\langle argument_1 \rangle$  相对  $\langle argument_2 \rangle$  从它的正常位置往右移动  $\langle number \rangle$  个数学单位。此命令常用于调整在其他重音之上的重音的位置。

```


$$\skew 2 \bar{\bar{z}} \quad \skew 3 \tilde{\tilde{y}} \quad \skew 4 \tilde{\hat{x}}$$


```



`\skewchar`  $\langle font \rangle$  [ $\langle number \rangle$  parameter]

字体的 `\skewchar` 是字体中的某个字符，它在字体度量文件中定义的紧排确定了数学重音的位置。也就是说，假设 T<sub>E</sub>X 要给字符 'x' 加上数学重音，则 T<sub>E</sub>X 检查字符对 '`x\skewchar`' 是否有个紧排；如果有，它就以该紧排的值移动该重音。T<sub>E</sub>X 放置数学重音的完整算法（这涉及到很多事情）在 *The T<sub>E</sub>Xbook* 附录 G 中描述。

如果 `\skewchar` 的值不在 0–255 的范围内，T<sub>E</sub>X 将紧排的值当作零。

注意  $\langle font \rangle$  是一个控制序列，它是字体的名称，而不是字体文件的名称  $\langle fontname \rangle$ 。小心：对 `\skewchar` 的赋值在编组结束时还原。如果你想局部改变 `\skewchar`，你需要显式地保存和还原它的原始值。

`\defaultskewchar` [ $\langle number \rangle$  parameter]

在执行 `\font` 命令读取字体的度量文件时，T<sub>E</sub>X 设定该字体的 `\skewchar` 等于 `\defaultskewchar`。如果 `\defaultskewchar` 的值不在 0–255 的范围内，T<sub>E</sub>X 默认就不设定 `\skewchar` 的值。Plain T<sub>E</sub>X 设定 `\defaultskewchar` 等于 -1，一般不需要改动它。

■

`\vcenter` {  $\langle vertical mode material \rangle$  }

`\vcenter to`  $\langle dimen \rangle$  {  $\langle vertical mode material \rangle$  }

`\vcenter spread`  $\langle dimen \rangle$  {  $\langle vertical mode material \rangle$  }

每个数学公式都有一个不可见的“轴线”，T<sub>E</sub>X 将它作为该公式的水平中心线。举个例子，由分式组成的公式的轴线就在分数线的中心。`\vcenter` 命令让 T<sub>E</sub>X 将  $\langle vertical mode material \rangle$  放入竖直盒子中，并将该竖直盒子与当前公式的轴线居中对齐。

此命令的第一种形式如上所述居中放置素材。后两种形式竖直扩展或收缩素材，如同 `\vbox` 命令（第 `\vbox` 页）。

```

 $\$ \{ n \}$ \choose k $\}$ \buildrel \rm \def \over \equiv $\>$
 $\vcenter{\hsize 1.5 in \noindent$ the number of
combinations of n things taken k at a time $\}$

```

$$\binom{n}{k} \stackrel{\text{def}}{=} \begin{array}{l} \text{the number of combina-} \\ \text{tions of } n \text{ things taken } k \\ \text{at a time} \end{array}$$



`\jot` [*<dimen>* parameter]

此参数定义为三个点的距离（除非你改变了它）。在用 `\openup` 命令分开陈列公式各行时，`\jot` 是一个实用的度量单位。<sup>2</sup>

```


$$\begin{aligned} & \text{\texttt{\$}\texttt{\$}\texttt{\backslash vbox{\texttt{\backslash halign{\texttt{\$}\texttt{\backslash hfil\#\texttt{\backslash hfil\$\texttt{\backslash cr x\texttt{\backslash cr y\texttt{\backslash cr}}}\texttt{\$}\texttt{\$}}}}}} \\ & \text{\texttt{\$}\texttt{\backslash openup2\texttt{\backslash jot}\texttt{\backslash vbox{\texttt{\backslash halign{\texttt{\$}\texttt{\backslash hfil\#\texttt{\backslash hfil\$\texttt{\backslash cr x\texttt{\backslash cr y\texttt{\backslash cr}}}\texttt{\$}\texttt{\$}}}}}}}} \end{aligned}$$


```

$$x$$

$$y$$

$$x$$

$$y$$

■

`\mkern` *<mudimen>*

此命令生成一个宽度为 *<mudimen>* 的紧排，即空白间隔。该紧排用数学单位表示，因此在不同样式中有不同的尺寸。除了使用数学单位外，此命令与水平模式的 `\kern`（第 [‘kern’](#) 页）的表现类似。

```


$$\text{\texttt{\$0}\texttt{\mkern13mu 0}\texttt{\quad}\texttt{\scriptscriptstyle 0}\texttt{\mkern13mu 0}\texttt{\texttt{\$}}}$$


```

0 0      o o

`\mskip` *<mudimen<sub>1</sub>>* `plus` *<mudimen<sub>2</sub>>* `minus` *<mudimen<sub>3</sub>>*

此命令生成一个水平粘连，它的自然宽度为 *<mudimen<sub>1</sub>>*，伸长度为 *<mudimen<sub>2</sub>>*，收缩量为 *<mudimen<sub>3</sub>>*。该粘连用数学单位表示，因此将随着样式的变化而变化。除了使用数学单位外，此命令与 `\hskip`（第 [‘hskip’](#) 页）的表现类似。

```


$$\text{\texttt{\$0}\texttt{\mskip 13mu 0}\texttt{\quad}\texttt{\scriptscriptstyle 0}\texttt{\mskip 13mu 0}\texttt{\texttt{\$}}}$$


```

---

<sup>2</sup> `\openup` [‘openup’](#)

0 0   o o

### `\nonscript`

在排版标号或小标号样式时，如果 `TEX` 在粘连或紧排跟前遇到此命令，它就丢弃该粘连或紧排。`\nonscript` 在其他样式中无任何效果。

此命令提供一种“收紧”标号和小标号样式中的间隔的方法；通常用小号字体排版这两个样式。在宏定义之外的地方，此命令很少用到。

```
\def\ab{a\nonscript\; b}
\ab^{\ab}
```

$a b^{ab}$

`\kern` (第 `'kern'` 页) 和 `\hskip` (第 `'hskip'` 页)。



### `\displaywidth`    [*dimen*] parameter]

此参数指定 `TEX` 对陈列公式所允许的最大宽度。如果 `TEX` 无法将陈列公式放入这样宽的空间中，它将生成一个过满的水平盒子并给出警告。`TEX` 在遇到 `'$$'` 开始陈列公式时就设定 `\displaywidth` 的值。它的初始值为 `\hsize` (第 `'hsize'` 页)，除非段落形状改变了。见 *The T<sub>E</sub>Xbook* 第 188--189 页中对此参数的更仔细的说明。

### `\displayindent`    [*dimen*] parameter]

此参数指定 `TEX` 对陈列公式的缩进量。`TEX` 在遇到 `'$$'` 开始陈列公式时就设定 `\displayindent` 的值。通常它的初始值为零，但如果段落形状表明该陈列公式需要移动距离  $s$ ，`TEX` 就设定 `\displayindent` 等于  $s$ 。见 *The T<sub>E</sub>Xbook* 第 188--189 页中对此参数的更仔细的介绍。

### `\predisplaysize`    [*dimen*] parameter]

`TEX` 设定此参数等于陈列公式之前的文本行的宽度。`TEX` 利用 `\predisplaysize` 确定是否让陈列公式的起始点位于前一行结尾处的左

边，即它在外观上是否可能与前一行重叠。如果会有重叠， $\text{\TeX}$  在排版陈列公式时使用 `\abovedisplayskip` 和 `\belowdisplayskip` 粘连；否则  $\text{\TeX}$  使用 `\abovedisplayshortskip` 和 `\belowdisplayshortskip` 粘连。见 *The  $\text{\TeX}$ book* 第 188--189 页中对此参数的更仔细的介绍。

`\abovedisplayskip`    [ *$\langle glue \rangle$*  parameter]

此命令指定当陈列公式的起始点位于前一行结尾处的左边时，即它在外观上可能与前一行有重叠时， $\text{\TeX}$  在陈列公式之前插入的竖直粘连的大小。Plain  $\text{\TeX}$  设定 `\abovedisplayskip` 等于 `12pt plus3pt minus9pt`。见 *The  $\text{\TeX}$ book* 第 188--189 页中对此参数的更仔细的介绍。

`\belowdisplayskip`    [ *$\langle glue \rangle$*  parameter]

此命令指定当陈列公式的起始点位于前一行结尾处的左边时，即它在外观上可能与前一行有重叠时， $\text{\TeX}$  在陈列公式之后插入的竖直粘连的大小。Plain  $\text{\TeX}$  设定 `\belowdisplayskip` 等于 `12pt plus3pt minus9pt`。见 *The  $\text{\TeX}$ book* 第 188--189 页中对此参数的更仔细的介绍。

`\abovedisplayshortskip`    [ *$\langle glue \rangle$*  parameter]

此命令指定当陈列公式的起始点位于前一行结尾处的右边时，即它在外观上不会与前一行有重叠时， $\text{\TeX}$  在陈列公式之前插入的竖直粘连的大小。Plain  $\text{\TeX}$  设定 `\abovedisplayshortskip` 等于 `0pt plus3pt`。见 *The  $\text{\TeX}$ book* 第 188--189 页中对此参数的更仔细的介绍。

`\belowdisplayshortskip`    [ *$\langle glue \rangle$*  parameter]

此命令指定当陈列公式的起始点位于前一行结尾处的右边时，即它在外观上不会与前一行有重叠时， $\text{\TeX}$  在陈列公式之后插入的竖直粘连的大小。Plain  $\text{\TeX}$  设定 `\belowdisplayshortskip` 等于 `7pt plus3pt minus4pt`。见 *The  $\text{\TeX}$ book* 第 188--189 页中对此参数的更仔细的介绍。

■

`\mathsurround` [  $\langle dimen \rangle$  parameter ]

此参数指定  $\text{\TeX}$  在文内数学公式（即放在两个 $\$$ 之间的公式）两边插入的间隔的大小。见 *The  $\text{\TeX}$ book* 第 162 页对此行为的进一步解释。Plain  $\text{\TeX}$  设定 `\mathsurround` 为 0pt。

`\nulldelimiterspace` [  $\langle dimen \rangle$  parameter ]

此参数指定空定界符生成的间隔的大小。Plain  $\text{\TeX}$  设定 `\nulldelimiterspace` 等于 1.2pt。

`\scriptspace` [  $\langle dimen \rangle$  parameter ]

此参数指定  $\text{\TeX}$  在上标或下标前后插入的间隔的大小。上标或下标之后的 `\nonscript` 命令（第 `\nonscript` 页）可以取消此间隔。Plain  $\text{\TeX}$  设定 `\scriptspace` 等于 0.5pt。

|                       |                         |
|-----------------------|-------------------------|
| <code>\mathord</code> | <code>\mathopen</code>  |
| <code>\mathop</code>  | <code>\mathclose</code> |
| <code>\mathbin</code> | <code>\mathpunct</code> |
| <code>\mathrel</code> |                         |

这些命令让  $\text{\TeX}$  把随后的结构归入指定的类（见 *The  $\text{\TeX}$ book* 第 154 页对类的定义）。它们按照类编号的大小顺序排列，从 0 到 6。它们主要用于按照指定的类调整该结构两边的间隔大小。

```
$\mathop{\rm minmax}\limits_{t \in A \cup B}\,t$
% By treating minmax as a math operator, we can get TeX to
% put something underneath it.
```

```
minmax t
t \in A \cup B
```

`\mathinner`

此命令让 T<sub>E</sub>X 将随后的结构视为“内部公式”，比如分式，并据此调整间隔。它与上面刚提到的类命令类似。

`\everymath` [ *(token list)* parameter ]

`\everydisplay` [ *(token list)* parameter ]

这两个命令分别指定 T<sub>E</sub>X 在每个文内公式或陈列公式开头插入的记号列。你可以利用 `\everymath` 或 `\everydisplay` 在每个数学公式开头作特殊处理。你务必清楚，若你需要同时处理两种公式，你必须设定这两个参数。

```
\everydisplay={\heartsuit\quad}
```

```
\everymath = {\clubsuit}
```

```
3 is greater than 2 for large values of 3.
```

```
$$4>3$$
```

```
♣3 is greater than ♣2 for large values of ♣3.
```

```
♥ 4 > 3
```









# 9



TeX ‘cmddesc’

```
\font
\font <control sequence> = <fontname>
\font <control sequence> = <fontname> scaled <number>
\font <control sequence> = <fontname> at <dimen>
```

```
 \font \font
 \font <fontname> \font
```

- 1) <control sequence> <fontname> (见后).
- 2) TeX <fontname> (.tfm).

cmr10 10

scaled <number> at <dimen>, ---设计大小的含意是, 字体在这个大小时, 表示最佳。否则将会载入一个缩放版的字体:

- 如果指定了 scaled <number>, 则字体将被放大 <number>/1000 倍。
- 如果指定了 at <dimen>, 则字体通过缩放大 <dimen>/ds 倍, 变为 <dimen> 大小, 其中, ds 是 <fontname> 的设计大小。<dimen> 和 ds 的单位往往使用点。

放大率可以小于 1, 这样做就会使字体缩小尺寸。

你往往需要为载入的字体的每一个使用的放大率提供一个字体轮廓文件 (第 ``shape'` 页). 当然, 一些设备驱动可以使用打印机内置的字体。这些字体不需要字体轮廓文件。

更多信息请参见 “” depth height (第 ``'` 页) 和 “” depth height (第 ``'` 页) .

```
\font\tenttt = cmtt10
\font\bigttfont = cmtt10 scaled \magstep2
\font\eleventtfont = cmtt10 at 11pt
First we use {\tenttt regular CM typewriter}.
Then we use {\eleventtfont eleven-point CM typewriter}.
Finally we use {\bigttfont big CM typewriter}.
```

First we use regular CM typewriter. Then we use eleven-point CM typewriter. Finally we use **big** CM typewriter.

```
\fontdimen <number> [<dimen> parameter]
 <fontname> <font-name> TEX
```

```
1
2
3
4
5 x (1ex)
6 m (1em)
7
```

T<sub>E</sub>X (见 `\spaceskip`, 第 ``spaceskip'` 页). (见 `\spaceskip`, 第 ``spaceskip'` 页). ■  
*The T<sub>E</sub>Xbook* 433 plain T<sub>E</sub>X 15

```

Here's a line printed normally.\par
\fontdimen2\font = 3\fontdimen2\font
% Triple the interword spacing.
\noindent Here's a really spaced-out line.

```

```

Here's a line printed normally.
Here's a really spaced-out line.

```

```

\magnification = <number>
\mag [<number> parameter]
 \magnification f ‘\magnification
 f <number> \hsize 6.5true in\vsiz 8.9true in 81/2 11 f
0 32768 f/1000 1000 1.2 \magstep \magstephalf
 \magnification \dimen0 = \magnification TEX
 \mag \mag \magnification \mag1

```

```

\magnification = \magstep2
% 1.44 (=1.2x1.2)

```

```

\magstep <number>
 true 1.2r r <number> <number> 0 52

```

```

\magstephalf
 true $\sqrt{1.2}$ 1 1.23

```

---

<sup>1</sup> \mag T<sub>E</sub>X \magnification plain T<sub>E</sub>X \magnification \mag \mag LaTeX  
\magnification \mag LaTeX

<sup>2</sup> plain T<sub>E</sub>X \magstep1 1200, \magstep2 为 1440, 等等。

\magnification = \magstep1 % 1.2

<sup>3</sup> plain T<sub>E</sub>X \magstephalf 1095。

\magnification = \magstephalf



`\number`  $\langle number \rangle$   
 $\langle number \rangle$   $\langle number \rangle$

`\number 24 \quad \count13 = -10000 \number\count13`

24 -10000

☆ `\romannumeral`  $\langle number \rangle$   
 $\langle number \rangle$   $\langle number \rangle$  `\romannumeral`

`\romannumeral 24 \quad (\romannumeral -16)\quad`  
`\count13 = 6000 \romannumeral\count13`

xxiv () mmmmmm



`\time` [  $\langle number \rangle$  parameter ]  
`TEX \time 720. TEX \time \time`

`\day` [  $\langle number \rangle$  parameter ]  
`TEX 1 31 \day` (见上面 `\time` 的说明).

`\month` [  $\langle number \rangle$  parameter ]  
`TEX 1 12 \time` (见上面 `\time` 的说明).

`\year` [*⟨number⟩* parameter]

`\TeX` () 1991 `\year` (见上面 `\time` 的说明).

`\fmtname`

`\fmtversion`

`\TeX` plain `\TeX` L<sup>A</sup>`\TeX`

`\fmtname\ ~\fmtversion`

=

eplain 1.9: 26 April 1991 (and plain 3.141592653)。

`\jobname`

`\TeX` hatter.tex, `\jobname` hatter. `\jobname`

`\newwrite\indexfile` `\openout\indexfile = \jobname.idx`  
`% 'hatter.tex' 'hatter.idx'.`

■

`\meaning` *⟨token⟩*

*⟨token⟩* the (第 `the` 页) `\TeX`

[`{\tt \meaning\ejct}`] [`\meaning\tenrm`] [`\meaning Y`]

[`macro:->\par \break`] [`select font cmr10`] [`the letter Y`]

`\string` *⟨control sequence⟩*

*⟨control sequence⟩* `\escapechar` `\TeX` 12



`\csname 'csname'`

`{\tt \string\bigbreak}`

`\bigbreak`

`\escapechar [ <number> parameter ]`

`TeX ASCII \string TeX 92 ASCII \escapechar 0–255 TeX`

`\escapechar = '!`

`{\tt \string\bigbreak}`

`!bigbreak`

`\fontname <font>`

`<font> <font> <fontname>.`

`\font\myfive=cmr5 [\fontname\myfive]`

`[cmr5]`

`\begingroup`

`\endgroup`

`\begingroup \endgroup`

`TeX \begingroup \endgroup \begingroup \endgroup \begingroup■`

`\endgroup`

```

\def\leftarrow{\begingroup\it 二}
\def\endleftarrow{\endgroup 四}
\leftarrow \endleftarrow

```

```

☆ {
 \bgroup
}
 \egroup

```

TeX `\bgroup \egroup \bgroup \egroup` ‘{’ ‘}’ 是等价的。

当你定义两个成对的宏时，`\bgroup` 和 `\egroup` 会非常有用。这两个宏中其中一个开始一个由花括号定界的结构（不一定是一个组），而另一个结束该结构。你不能使用常规的花括号来定义这样的宏，否则你的宏定义会包含没有匹配的花括号，而这样的定义是不被 TeX 所接受的。一般情况下你需要在你没有办法使用花括号时使用这些命令。

```
{\it \/}
```

```

\def\leftarrow{\vbox\bgroup}
% 在这里你不能使用一个 { 来代替 \bgroup,
% 否则 TeX 不会识别这个宏
\def\endleftarrow{{\#1\egroup} 二}
% , \egroup ,
% \bgroup .
% !
\leftarrow \endleftarrow{\hrule width 1in}

```

---

```

\global
 ‘ \global

```

```

{\global\let\leftarrow = \leftarrow}
$a \leftarrow b$

```

$$a \leftarrow b$$

`\globaldefs` [*<number>* parameter]

`TEX`

- `\globaldefs` (默认值) `\global` (`\gdef` `\xdef` (第 `\gdef` 页) `\global` ).
- `\globaldefs` `global`.
- `\globaldefs` `\global`

`\aftergroup` *<token>*

`TEX` *<token>* *<token>* `\aftergroup`  
`\aftergroup`

`\def\neg{负}` `\def\pos{正}`

% `\aftergroup` ,

% , ,

% .

`\def\arith#1{是否 $#1>0$? \begingroup`

`\ifnum #1>-1 是 \aftergroup\pos`

`\else \aftergroup\neg\fi`

`, \endgroup. }`

`\arith 2`

`\arith {-1}`

`2 > 0? , . -1 > 0? , .`

`\afterassignment` *<token>*

`TEX` *<token>* *<token>* `\afterassignment` `\afterassignment`

`\afterassignment` *The T<sub>E</sub>Xbook* page 279.

```

\def\setme{\afterassignment\setmeA\count255}
\def\setmeA{${\number\count255\advance\count255 by 10
+10=\number\count255$}
\setme = 27
% \setme , TeX \count255
% 27, \setmeA.

```

$27 + 10 = 37$

■

```

\def <control sequence> <parameter text> { <replacement text> }
 <parameter text> <replacement text> <control sequence> ‘

```

```

\def\add#1+#2=?{#1+#2&=
 \count255=#1 \advance\count255 by #2 \number\count255\cr}■
$$\eqalign{
 \add 27+9=?
 \add -5+-8=?}$$

```

$27 + 9 = 36$

$-5 + -8 = -13$

```

\edef <control sequence> <parameter text> { <replacement text> }
 \def TeX \edef <replacement text> <replacement text> TeX
\edef <replacement text> \hbox The TeXbook 212-215
 \noexpand (第 `noexpand' 页) \expandafter (第 `expandafter' 页)■

```

```
\write,\message,\errmessage,\wlog \csname \edef
```

```
\def\aa{xy} \count255 = 1
\edef\bb{w\ifnum \count255 > 0\aa\fi z}
% \def\bb{wxyz}
\def\aa{} \count255 = 0 % \bb
\bb
```

```
wxyz
```

```
\gdef <control sequence> <parameter text> { <replacement text> }
\global\def.
```

```
\xdef <control sequence> <parameter text> { <replacement text> }
\global\xdef.
```

```
\long
```

```
TeX \par (第`@par' 页), TeX \long \par TeX \long
```

```
\long\def\aa#1{\par\hrule\smallskip#1\par\smallskip\hrule}
\aa{这是第一行。 \par
}
% \long, TeX
```

---



---

```
\outer
```

```
TeX “ TeX
```

```
\outer\def\chapterhead#1{%
\eject\topglue 2in \centerline{\bf #1}\bigskip}
% \chapterhead
% .
```

```
\chardef <control sequence>=<charcode>
 <charcode> <control sequence>. \chardef 0-255
```

```
\chardef\percent = '\% 21\percent, {\it 19\percent}
%
```

21%, 19%

```
\mathchardef <control sequence>=<mathcode>
 <control sequence>
```

```
\mathchardef\alphachar = "010B % plain TeX
\alphachar
```

$\alpha$

■

```
\let <control sequence> = <token>
 <control sequence> <token> <token> <control sequence> <token>
```

```
\futurelet <control sequence> <token1> <token2>
 TEX <control sequence> <token2> \let <token1> <token2> \futurelet
TEX
```

```
\def\predict#1{\toks0={#1}\futurelet\next\printer}
% \next \predict .
\def\printer#1{一个 \punc\ 放在 \the\toks0 前。}
\def\punc{%
 \ifx\next; 分号 \else
 \ifx\next, 逗号 \else
 ‘\next’\fi\fi}
\predict{三月}; \predict{四月}, \predict{七月}/
```

“/”

```
\csname <token list> \endcsname
 <token list> <token list> \csname \edef 'edef' TEX \csname
\string 'string'
```

```
\def\capTe{Te}
 \csname\capTe X\endcsname.
```

T<sub>E</sub>X.

■

```
\expandafter <token1> <token2>
 TEX <token2> <token1> <token2>, '{' \string
```

```
\def\aa{xyz}
\tt % Use this font so '\' prints that way.
[\string\aa] [\expandafter\string\aa]
[\expandafter\string\csname TeX\endcsname]
```

```
[\aa] [xyz] [\TeX]
```

```
\noexpand <token>
 TEX <token> <token> TEX \noexpand, <token>
```

```

\def\bunny{兔子}
\edef\magic{把 \noexpand\bunny\ 从帽子中取出! }
% \noexpand, \bunny ‘兔子’
\let\oldbunny=\bunny \def\bunny{兔} \magic
\let\bunny=\oldbunny \magic

```

```

\the <token>
 <token> <token>
 ■ TEX, \parindent \deadcycles
 ■ , \count0
 ■ \catcode‘(
 ■ \fontdimen3\sevenbf
 ■ \hyphenchar \skewchar, \skewchar\teni
 ■ \lastpenalty, \lastskip, \lastkern (当前的水平或竖直列表的
 最后一项的数值).
 ■ \chardef \mathchardef

\the
 ■ \the ,
 ■ \the <token variable>, \the\everypar
 The TEXbook 214–215 \the

\the\vsizel =
‘(’ \the\catcode ‘(.

548.4975pt。 ‘(字符的类码是 12。

‘convert’, \showthe (第 ‘showthe’ 页).

```

■

```

\if <token1> <token2>
 <token1> <token2> TEX \if <token1> <token2> \let TEX 256

```



```

\def\first{abc}
\if\first true\else false\fi;
% ‘‘c’’ is left over from the expansion of \first.
% It lands in the unexecuted ‘‘true’’ part.
\if a\first\ true\else false\fi;
% Here ‘‘bc’’ is left over from the expansion of \first
\if \hbox\relax true\else false\fi
% Unexpandable control sequences test equal with ‘‘if’’

```

false; bc true; true

```

\ifcat <token1> <token2>
 <token1> <token2> TEX \ifcat <token1> <token2> \let TEX 16

```

```

\ifcat axtrue\else false\fi;
\ifcat]}true\else false\fi;
\ifcat \hbox\day true\else false\fi;
\def\first{12345}
\ifcat (\first true\else false\fi
% ‘‘2345’’ lands in the true branch of the test

```

true; false; true; 2345true

```

\ifx <token1> <token2>
 <token1> <token2> \if \ifcat \ifx \ifx <token1> <token2>
\ifx TEX

```

1)

2)

a)

b) T<sub>E</sub>X

3)

a) “第一层”展开即它们的替换文本相同；

b) 它们有相同的 \long (第 `\long` 页) 和 \outer (第 `\outer` 页) 状态。

`\if`

```
\ifx\alice\rabbit true\else false\fi;
% true since neither \rabbit nor \alice is defined
\def\aa{a}%
\ifx \aa true\else false\fi;
% false since one token is a macro and the other isn't
\def\first{\a}\def\second{\aa}\def\aa{a}%
\ifx \first\second true\else false\fi;
% false since top level expansions aren't the same
\def\third#1:{(#1)}\def\fourth#1?{(#1)}%
\ifx\third\fourth true\else false\fi
% false since parameter texts differ
```

true; false; false; false

```
\ifnum <number1> <relation> <number2>
 <number1> <number2> <relation> '<', '=' '>' 127 \pageno
\count22 \hbadness TEX \ifnum <number1> <relation> <number2>
<number2>
```

```
\count255 = 19 \ifnum \count255 > 12 true\else false\fi
```

true

```
\ifodd <number>
 <number> TEX \ifodd <number> <number>
```

```
\count255 = 19
\ifodd 5 true\else false\fi
```

true

```
\ifdim <dimen1> <relation> <dimen2>
```

```

 <dimen1> <dimen2> <relation> '<' '=' '>' 1in \dimen6 \parindent
TeX \ifdim <dimen1> <relation> <dimen2> <dimen2>

```

```

\dimen0 = 1000pt \ifdim \dimen0 > 3in true\else false\fi

```

```

true

```

```

\ifhmode

```

```

\ifvmode

```

```

\ifmmode

```

```

\ifinner

```

```

 TEX

```

```

 ▪ \ifhmode TEX

```

```

 ▪ \ifvmode TEX

```

```

 ▪ \ifmmode TEX

```

```

 ▪ \ifinner TEX “内部” 模式中： 即位于受限水平模式、内部竖直模
 式或者文内数学模式中。

```

```

\def\modes{\bf

```

```

 \ifhmode

```

```

 \ifinner IH\else H\fi

```

```

 \else\ifvmode

```

```

 \ifinner \hbox{IV}\else \hbox{V}\fi

```

```

 \else\ifmmode \hbox{M}\else

```

```

 error\fi\fi\fi}}

```

```

Formula \modes; then \modes,

```

```

 \hbox{next \modes\ and \vbox{\modes}}.

```

```

\par\modes

```

```

Formula M; then H, next IH and IV.

```

```

V

```

```

\ifhbox <register>

```

```

\ifvbox <register>

```

```

\ifvoid <register>

```

$\langle register \rangle$   $\langle register \rangle$   $n$

- `\ifhbox \box  $n$`
- `\ifvbox \box  $n$`
- `\ifvoid \box  $n$`

```
\setbox0 = \vbox{} % empty but not void
\setbox1 = \hbox{a}
\setbox2 = \box1 % makes box1 void
\ifvbox0 true\else false\fi;
\ifhbox2 true\else false\fi;
\ifvoid1 true\else false\fi
```

```
true; true; true
```

`\ifeof  $\langle number \rangle$`   
 $\langle number \rangle$

`\ifcase  $\langle number \rangle$  $\langle case_0 text \rangle$  \or  $\langle case_1 text \rangle$  \or ... \or  $\langle case_n text \rangle$`   
`\else  $\langle otherwise text \rangle$  \fi`  
 $\langle number \rangle$   $k$   $\langle case_k text \rangle$   $\text{\TeX}$   $\langle otherwise text \rangle$  \else  $\text{\TeX}$

```
\def\whichday#1{\ifcase #1<day 0>\or Sunday\or Monday%
\or Tuesday\or Wednesday\or Thursday\or Friday%
\or Saturday\else Nonday\fi
\ is day \##1. }
\whichday2 \whichday3 \whichday9
```

Monday is day #2. Tuesday is day #3. Nonday is day #9.

```
\iftrue
\iffalse
```

```

\def\isbigger{\let\bigger=\iftrue}
\def\isnotbigger{\let\bigger=\iffalse}
% These \let's MUST be buried in macros! If they aren't,
% TeX erroneously tries to match them with \fi.
\def\test#1#2{\ifnum #1>#2 \isbigger\else\isnotbigger\fi}
\test{3}{6}
\bigger$3>6$\else$3\le6$\fi

```

$3 \leq 6$

\else

“假”时的另一选择。

\fi

```

\newif \if<test name>
 <test name> alpha \alphatrue\alphafalse \ifalpha /
 ▪ \alphatrue alpha
 ▪ \alphafalse alpha
 ▪ \ifalpha alpha
 alpha TEX \newif\ifalpha
 \newif

```

```

\newif\iflong \longtrue
\iflong Rabbits have long ears.
\else Rabbits don't have long ears.\fi

```

Rabbits have long ears.

■

```

\loop α \ifΩ β \repeat
\repeat
 TEX α β \ifΩ 'conds' \repeat \fi \fi \else \newif

```

$\text{\TeX}$  \loop

- 1)  $\alpha$
- 2)  $\text{\if}\Omega$
- 3)  $\beta$
- 4)

```
\count255 = 6
\loop
 \number\count255\
 \ifnum\count255 > 0
 \advance\count255 by -1
 \repeat
```

6 5 4 3 2 1 0

```
\newif\ifnotdone % \newif uses \count255 in its definition
\count255=6
\loop
 \number\count255\
 \ifnum\count255 < 1 \notdonefalse\else\notdonetrue\fi
 \ifnotdone
 \advance\count255 by -1
 \repeat
```

6 5 4 3 2 1 0

■

\relax

$\text{\TeX}$

```
\def\medspace{\hskip 12pt\relax}
% The \relax guards against the possibility that
% The next tokens are ‘plus’ or ‘minus’.
```

```
\empty
\relax
```

■

```
\count <register> = <number> \count <register>
\dimen <register> = <dimen> \dimen <register>
\skip <register> = <glue> \skip <register>
\muskip <register> = <muglue> \muskip <register>
\toks <register> = <token variable> \toks <register>
\toks <register> = { <token list> }
```

```
‘=’ \the
\newcount ‘@newcount’
\count 4TeX 0–9 The TeXbook 119 \count255
```

```
\count255 = 17 \number\count255
```

```
17
```

```
\dimen \dimen0 \dimen9 \dimen255
```

```
\dimen0 = 2.5in
\hbox to \dimen0{${\Leftarrow}\hfil${\Rightarrow}}
```

```

<=>
|-----| 3 in
4 TeX
```

```
\skip \dimen \skip0 \skip9 \skip255
```

```
\skip2 = 2in
\Rightarrow\hskip \skip2 \Leftarrow
```

```
⇒
|-----| 2in
⇐
```

```
\muskip \skip mu “ mu \muskip0 \muskip9 \muskip255
```

```
\muskip0 = 24mu % An em and a half, no stretch or shrink.
$\mathop{a \mskip\muskip0 b}\limits^{\{a \mskip\muskip0 b\}}$
% Note the difference in spacing.
```

```

a b
a b
```

```
\toks
 \the TEX TEX
```

```
\toks0 = {the \oystereaters\ were at the seashore}
% This assignment doesn't expand \oystereaters.
\def\oystereaters{Walrus and Carpenter}
\toks1 = \toks0
% the same tokens are now in \toks0 and \toks1
Alice inquired as to whether \the\toks1.
```

Alice inquired as to whether the Walrus and Carpenter were at the seashore.

```
\maxdimen
 TEX ⟨dimen⟩ 18
```

```
\maxdepth = \maxdimen % Remove restrictions on \maxdepth.
```

`\advance` (第 `'advance'` 页), `\multiply`, `\divide` (第 `'divide'` 页),  
`\setbox`, `\box` (第 `'box'` 页).





|                         |                           |
|-------------------------|---------------------------|
| <code>\newcount</code>  | <code>\newread</code>     |
| <code>\newdimen</code>  | <code>\newwrite</code>    |
| <code>\newskip</code>   | <code>\newfam</code>      |
| <code>\newmuskip</code> | <code>\newinsert</code>   |
| <code>\newtoks</code>   | <code>\newlanguage</code> |
| <code>\newbox</code>    |                           |

- `\newcount\newdimen\newskip\newmuskip\newtoks \newbox`
- `\newread \newwrite`
- `\newfam`
- `\newinsert`
- `\newlanguage`

- `\newcount\newdimen\newskip\newmuskip \newtoks`  
`\newdimen\listdimen`  
`\listdimen`
- `\newbox\newread\newwrite\newfam\newinsert \newlanguage`  
`\newbox\figbox`  
`\figbox \box`  
`\setbox\figbox = \vbox{...}`

```

\countdef <control sequence> = <register>
\dimendef <control sequence> = <register>
\skipdef <control sequence> = <register>
\muskipdef <control sequence> = <register>
\toksdef <control sequence> = <register>
 <control sequence> <register> \newcount

```

```
\countdef\hatters = 19 % \hatters now refers to \count19
\toksdef\hares = 200 % \hares now refers to \toks200
```


\newif (第`@newif' 页), \newhelp (第`@newhelp' 页).

■

```
\advance <count register> by <number>
\advance <dimen register> by <dimen>
\advance <skip register> by <glue>
\advance <muskip register> by <muglue>
 <glue> <muglue> <glue> <dimen> <dimen> scaled point
<number> “ by TEX
```

```
\count0 = 18 \advance\count0 by -1 \number\count0\par
\skip0 = .5in \advance\skip0 by 0in plus 1in % add stretch
\hbox to 2in{a\hskip\skip0 b}
```

17



```
\multiply <register> by <number>
\divide <register> by <number>
 <register> <number> \count\dimen\skip \muskip \skip \muskip
‘skip’ by TEX
 <dimen> <number> -2.5\dimen2 <glue> <dimen> <glue>
2\baselineskip <dimen> \baselineskip
```

```

\count0 = 9\multiply \count0 by 8 \number\count0 ;
\divide \count0 by 12 \number\count0 \par
\skip0 = 20pt plus 2pt minus 3pt \multiply \skip0 by 3
Multiplied value of skip0 is \the\skip0.\par
\dimen0 = .5in \multiply\dimen0 by 6
\hbox to \dimen0{a\hfil b}

```

72; 6

Multiplied value of skip0 is 60.0pt plus 6.0pt minus 9.0pt.

a b



☆ \bye

TEX

\end

TEX \bye \end



☆ \input <filename>

TEX <filename> TEX

\input \input

```

 TEX ‘ TEX TEX .tex

\input macros.tex
\input chap1 % equivalent to chap1.tex

\endinput
 TEX

\inputlineno

\openin <number> = <filename>
 TEX <filename> <number> <number> 0 15 \read

 \newread ‘@newread’ \openin

\newread\auxfile \openin\auxfile = addenda.aux
% \auxfile now denotes the number of this opening
% of addenda.aux.

\closein <number>
 TEX <number>

\closein\auxfile

\read <number> to <control sequence>
 TEX <number> <control sequence> TEX TEX TEX
 <number> 15 \openin TEX ‘<control sequence>=’, 等待你键入一
 行输入，然后将该行输入赋予 <control sequence>。如果 <number> 小于
 零，它同样从终端读取输入，但不显示提示符。

\read\auxfile to \holder
% Expanding \holder will produce the line just read.

```

```

\openout <number> = <filename>
 TEX <filename> <number> <number> 0 15 \write
\openout TEX .dvi \openout \openout \immediate
TEX
\newwrite '@newwrite' \openout

\newwrite\auxfile \openout\auxfile = addenda.aux
% \auxfile now denotes the number of this opening
% of addenda.aux.

\closeout <number>
 TEX <number>
\closeout TEX .dvi \closeout \closeout \immediate

\closeout\auxfile

\write <number> { <token list> }
 TEX <token list> <number> TEX .dvi \write \immediate
 immediate) 的 \write, TEX 直到实际写入文件时才展开 <token
 list> 中的宏。宏展开遵循与 \edef (第 'edef' 页) 一样的规则。任何
 不是宏名称的控制序列被写成 \escapechar 加上控制序列名, 后面再加
 一个空格。<token list> 中的任何 '#' 记号被重复两遍, 即写成 '##'。
 如果 <number> 不在 0 到 15 的范围中, TEX 将 <token list> 写入到
 日志文件。如果 <number> 大于 15 或该输出流未关联文件, TEX 还同
 时将 <token list> 写到终端中。

\def\aa{a a}
\write\auxfile{\hbox{$x#y$} \aa}
% Writes the string '\hbox {$x##y$} a a' to \auxfile.

```

```

\immediate
 \openout\closeout \write TEX

\immediate\write 16{I'm stuck!}
% has the same effect as \message

\special { <token list> }
 TEX <token list> .dvi \special \special <token list>
\special \special

\special{graphic expic}
% Display the graphics file 'expic' here.

\newlinechar [<number> parameter]
 \write\message \errmessage TEX \newlinechar 0-255
Plain TEX \newlinechar -1

\newlinechar = '\^J
\message{This message appears^^Jon two lines.}

This message appears
on two lines.

\newread, \newwrite (第`@newwrite' 页).

```



```

\catcode <charcode> [<number> table entry]

```

ASCII  $\langle charcode \rangle$  ‘catcodes’ T<sub>E</sub>X

```
\catcode ‘\[= 1 \catcode ‘\] = 2
% Make [and] act like left and right braces.
```

\active

13

```
\catcode ‘\@ = \active % Make @ an active character.
```

\mathcode  $\langle charcode \rangle$  [  $\langle number \rangle$  table entry ]

ASCII  $\langle charcode \rangle$  ‘

```
\mathcode\> = "313E % as in plain TeX
% The > character has class 3 (relation), family 1 (math
% italic), and character code "3E
```

\delcode  $\langle charcode \rangle$  [  $\langle number \rangle$  table entry ]

ASCII  $\langle charcode \rangle$  T<sub>E</sub>X

$\langle number \rangle$   $\langle number \rangle$   $s_1 s_2 s_3 l_1 l_2 l_3$   $s_1 s_2 s_3 l_1 l_2 l_3$   $s_1 s_2 s_3$   $s_1$   $s_2 s_3$   
 $l_1 l_2 l_3$  \mathcode ‘mathcode’ \mathcode

```
\delcode ‘(= "028300 % As in plain TeX.
```

\endlinechar [  $\langle number \rangle$  parameter ]

T<sub>E</sub>X 0–255 Plain T<sub>E</sub>X \endlinechar ‘\^M  $\langle return \rangle$  ASCII

\ignorespaces

T<sub>E</sub>X \ignorespaces \ignorespaces \par

```
\def\aa#1{yes #1\ignorespaces}
\aa{may}
be
```

```
yes maybe
```

**T<sub>E</sub>X**

```
\errorstopmode
TEX TEX
```

```
\scrollmode
TEX ‘S’ ‘s’
```

```
\nonstopmode
TEX ‘R’ ‘r’
```

```
\batchmode
TEX ‘Q’ ‘q’
```

```
\pausing [⟨number⟩ parameter]
TEX TEX ⟨return⟩TEX
\pausing 1 TEX \show
```





```
\show <token>
```

```
\showthe <argument>
```

```
\showbox <number>
```

```
\showlists
```

```
TeX
```

```
▪ \show <token>
```

```
▪ \showthe \the <argument> 'the'
```

```
▪ \showbox <number>
```

```
▪ \showlists TeX The TeXbook 88-89 \showlists
```

```
\show \showtheTeX \showbox \showlists TeX \tracingonline 'tracingonline' \trac
```

```
\show TeX TeX <return>
```

```
\showbox \showboxbreadth \showboxdepth 'showboxbreadth'
```

```
5 3 '..etc.'
```

```
\show a
```

```
\show \hbox
```

```
\show \medskip
```

```
\show &
```

```
> the letter a.
```

```
> \hbox=\hbox.
```

```
> \medskip=macro:
```

```
->\vskip \medskipamount .
```

```
> alignment tab character &.
```

```
\showthe\medskipamount
\toks27={\hbox{Joe's\quad\ Diner}}
\showthe\toks27
```

```
> 6.0pt plus 2.0pt minus 2.0pt.
> \hbox {Joe's\quad \ Diner}.
```

```
\setbox 3=\vbox{\hbox{A red dog.}\hrule A black cat.}
\showbox 3
```

```
> \box3=
\vbox(16.23332+0.0)x53.05565
.\hbox(6.94444+1.94444)x46.41675
..\tenrm A
..\glue 3.33333 plus 1.66498 minus 1.11221
..\tenrm r
..\tenrm e
..\tenrm d
..etc.
.\rule(0.4+0.0)x*
.\hbox(6.94444+0.0)x53.05565
..\tenrm A
..\glue 3.33333 plus 1.66498 minus 1.11221
..\tenrm b
..\tenrm l
..\tenrm a
..etc.
```

```
\vbox{A \hbox
 {formula
 $x \over y\showlists$}}
```

```

math mode entered at line 3
\mathord
.\fam1 y
this will be denominator of:
\fraction, thickness = default
\\mathord
\.\fam1 x
restricted horizontal mode entered at line 2
\tenrm f
\tenrm o
\tenrm r
\tenrm m
\kern-0.27779
\tenrm u
\tenrm l
\tenrm a
\glue 3.33333 plus 1.66666 minus 1.11111
spacefactor 1000
horizontal mode entered at line 1
\hbox(0.0+0.0)x20.0
\tenrm A
\glue 3.33333 plus 1.66498 minus 1.11221
spacefactor 999
internal vertical mode entered at line 1
prevdepth ignored
vertical mode entered at line 0
prevdepth ignored

```

```
\showboxbreadth\showboxdepth 'showboxbreadth'
```

■

```

\tracingonline [<number> parameter]
 TEX \showbox \showlists

```

```
\tracingcommands [<number> parameter]
 1TEX \tracingonline \tracingcommands 2 TEX
```

```
\tracingcommands = 1 If $x+y>0$ we quit.\par
On the other hand, \tracingcommands = 0
```

```
{vertical mode: the letter I}
{horizontal mode: the letter I}
{blank space }
{math shift character $}
{math mode: the letter x}
{the character +}
{the letter y}
{the character >}
{the character 0}
{math shift character $}
{horizontal mode: blank space }
{the letter w}
{blank space }
{the letter q}
{blank space }
{\par}
{vertical mode: the letter 0}
{horizontal mode: the letter 0}
{blank space }
{the letter t}
{blank space }
{the letter o}
{blank space }
{the letter h}
{blank space }
{\tracingcommands}
```

```
\tracinglostchars [<number> parameter]
 TEX \tracingonline Plain TEX 1
```

```
\tracinglostchars = 1
A {\nullfont few} characters.
```

```
Missing character: There is no f in font nullfont!
Missing character: There is no e in font nullfont!
Missing character: There is no w in font nullfont!
```

```
\tracingmacros [<number> parameter]
1 TEX \tracingmacros 2 TEX \output \everycr \tracingonline■
```

```
\def\first \b, then \c}
\def\b{b} \def\c{c}
\tracingmacros = 2
Call \a once.
```

```
\a ->first \b , then \c
```

```
\b ->b
```

```
\c ->c
```

```
\tracingoutput [<number> parameter]
TEX .dvi \tracingonline \showboxbreadth \showboxdepth 'showboxbreadth'■
\tracingoutput 1
```

```
% This is the entire file.
\tracingoutput = 1 \nopagenumbers
One-line page. \bye

Completed box being shipped out [1]
\ vbox(667.20255+0.0)x469.75499
.\ vbox(0.0+0.0)x469.75499, glue set 13.99998fil
..\ glue -22.5
..\ hbox(8.5+0.0)x469.75499, glue set 469.75499fil
...\ vbox(8.5+0.0)x0.0
...\ glue 0.0 plus 1.0fil
..\ glue 0.0 plus 1.0fil minus 1.0fil
.\ vbox(643.20255+0.0)x469.75499, glue set 631.2581fill
..\ glue(\topskip) 3.05556
..\ hbox(6.94444+1.94444)x469.75499, glue set 386.9771fil
...\ hbox(0.0+0.0)x20.0
...\ \tenrm 0
...\ \tenrm n
...\ \tenrm e
...\ \tenrm -
...etc.
..\ glue 0.0 plus 1.0fil
..\ glue 0.0 plus 1.0fill
.\ glue(\baselineskip) 24.0
.\ hbox(0.0+0.0)x469.75499, glue set 469.75499fil
..\ glue 0.0 plus 1.0fil
```

```
\tracingpages [⟨number⟩ parameter]
TeX \tracingonline TeX The TeXbook 112–114
TeX \tracingpages TeX
```

```
\tracingparagraphs [⟨number⟩ parameter]
TeX \tracingonline TeX The TeXbook 98–99
TeX \tracingparagraphs TeX
```

```
\tracingrestores [<number> parameter]
 TEX \tracingonline
 TEX \tracingrestores TEX
```

```
\tracingstats [<number> parameter]
 1 TEX The TEXbook 300 \tracingstats 2 TEX \shipout ‘shipout’ ■
\tracingonline TEX \tracingstats
 TEX \tracingstats TEX
 TEX TEX
```

```
\tracingstats=1
```

Here is how much of TeX’s memory you used:

```
4 strings out of 5540
60 string characters out of 72328
5956 words of memory out of 262141
921 multiletter control sequences out of 9500
14794 words of font info for 50 fonts, out of 72000 for 255
14 hyphenation exceptions out of 607
7i,4n,1p,68b,22s stack positions out of 300i,40n,60p,3000b,4000s
```

```
\tracingall
 TEX \tracingonline 1
```

```
\showboxbreadth [<number> parameter]
 TEX \showbox \tracingoutput Plain TEX \showboxbreadth 5
```

```
\showboxdepth [<number> parameter]
 TEX \showbox \tracingoutput Plain TEX \showboxdepth 3
```

■

```
\message { <token list> }
\errmessage { <token list> }
 <token list> TEX \edef ‘edef’
```

```
\errmessageTEX \errhelp
\newlinechar ‘newlinechar’
```

```
\message{Starting a new section.}
```

```
\wlog { <token list> }
 <token list> TEX \edef ‘edef’ <token list>
```

```
\wlog{Take two aspirins and call me in the morning.}
```

Take two aspirins and call me in the morning.

```
\errhelp [<token list> parameter]
 TEX \errmessage \errmessage \errhelp \newhelp \newlinechar■
```

```
\newhelp <control sequence> { <help text> }
 <help text> <control sequence> \errmessage <control sequence>
\errhelp ‘H’ ‘h’, TEX 将显示出帮助文本。
```

```
\newhelp\pain{Your input includes a token that I find^^J
to be offensive. Don’t bother me again with this^^J
document until you’ve removed it.}
\errhelp = \pain \newlinechar = ‘\^^J
% ^^J will start a new line
\errmessage{I do not appreciate receiving this token}
```

```
! I do not appreciate receiving this token.
1.8 ...t appreciate receiving this token.}
```

? H

```
\Your input includes a token that I find
to be offensive. Don’t bother me again with this
document until you’ve removed it.
```



`\errorcontextlines` [ *<number>* parameter ]

T<sub>E</sub>X 0

I\errorcontextlines=100\oops

Plain T<sub>E</sub>X \errorcontextlines 5

`\write` *'write'* `\escapechar` *'escapechar'*

T<sub>E</sub>X

`\dump`

T<sub>E</sub>X ‘<sup>5</sup> virtex T<sub>E</sub>X <sup>5</sup> T<sub>E</sub>X \dump \dump initex T<sub>E</sub>X T<sub>E</sub>X

`\everyjob` [ *<token list>* parameter ]

T<sub>E</sub>X \everyjob

---

<sup>5</sup>virtex initex T<sub>E</sub>X initex virtex tex tex







TeX

TeX

- 
- 

```
TeX \vbox 'vbox' \par TeX
\interlinepenalty 'interlinepenalty' 10000 TeX TeX
\nobreak (第 'vnobreak' 页) TeX 10000 \nobreak \par TeX
\nobreak \nobreak TeX \nobreak
\eject 'eject' TeX '\vadjust{\vfill\eject}' 'vadjust' \eject
\vfill 'vfill' TeX \eject
```

\eject \vfill TeX an underfull \vbox (badness 10000) has occurred while \output is active.”上面讲的封装素材在一起的任一方法中，类似的问题也会出现。

\filbreak 命令（第 'filbreak' 页）提供了一种方式来把一个自然段或几个段落（或者其他竖直模式素材）放在同一页面上。当你用 \filbreak 来封闭一个自然段的时候，如果段落内容可以放置在一页，TeX 会有效地忽略那些 \filbreak；如果不是，TeX \filbreak

`\filbreak`

`\filbreak`

`\par`

`\filbreak`

`\par`

`\filbreak`

`\vdots`

`\par`

`\filbreak`

```

TEX \filbreak TEX
 \looseness 'looseness' \looseness TEX \looseness
\tracingpages 'tracingpages' 1

```

```

\adjust{\vfill\eject}\break

```

```

TEX \looseness 'looseness' \parskip (第 'parskip' 页) \baselineskip
'baselineskip'

```

```

\vskip 'vskip' TEX \topglue 'topglue'

```

T<sub>E</sub>X

- \break 'hbreak'
- (~) (见第 '@not' 页)，来防止在它们之间断开。
- 你可以在一些单词里插入一个或者多个自定连字符，来告诉 T<sub>E</sub>X 新的断词规则（参考 \-, depth height plus 0.1em minus 0.1 em 第 '@minus' 页）。
- 你可以使用 \hbox 命令（第 'hbox' 页），把几个单词封装在一个水平盒子里。

T<sub>E</sub>X T<sub>E</sub>X T<sub>E</sub>X

T<sub>E</sub>X T<sub>E</sub>X

- T<sub>E</sub>X \emergencystretch ‘emergencystretch’ T<sub>E</sub>X \tolerance ‘tolerance’ 10000 T<sub>E</sub>X \hfuzz ‘hfuzz’ T<sub>E</sub>X
- T<sub>E</sub>X T<sub>E</sub>X \raggedright ‘raggedright’

▪

Underfull \vbox (badness 10000) has occurred  
while \output is active

T<sub>E</sub>X T<sub>E</sub>X

T<sub>E</sub>X \vsize ‘vsize’ \baselineskip ‘baselineskip’  
\parskip ‘parskip’ \vfuzz ‘vfuzz’

▪

Overfull \vbox (296.30745pt too high) has occurred  
while \output is active

- \hbox \vbox ‘hbox’ ‘vbox’ \hss \vss ‘hss’
- \hbox \vbox ‘hbox’ ‘vbox’ \hfil \vfil ‘hfil’

T<sub>E</sub>X (\\_)

### T<sub>E</sub>X

‘%’。‘%’用于注释开始，但不一定真的要有注释内容。

宏定义如果写的不仔细，也可能导致多余的空白。当你应用一个宏，但是有多余的空白，那就要检查其定义，确保大括号后没有额外的空格以及紧接着括号之后，宏定义没有换行。人们经常在大括号后对宏定义换行，从而容易阅读。安全起见，应该在所有大括号之后要换行的地方放一个‘%’。可能有些地方不需要，但是这么做起码没有坏处。<sup>1</sup>

当你不能准确在源文档中定位多余的空白的时候，可以尝试设置 `\tracingcommands` (第 `\tracingcommands` 页) 为 2。在日志文件里，每一处空格都会输出一个 `{blank space}` 命令。

知道 T<sub>E</sub>X 关于空格的规则会有帮助。

- 1) 输入行开始之前的空格将被忽略。
- 2) 输入行结束之后的空格在任何情况下都被忽略。虽然行尾本身就被当作空格。(一个空白行产生的是 `\par` 标记。)
- 3) 多于一个的空格，如果在一起，将被处理为一个空格。因此，宏调用的参量之后的空格将不会和宏定义末尾的空格合并。你得到的是两个空格。
- 4) 控制字符之后的空格将被忽略。
- 5) 数字，尺寸以及粘连设定中的 `\plus` 和 `\minus` 之后的空格在执行中也被忽略。<sup>2</sup>

如果你改变了空格符或行尾符的类别码，这些规则就不算数了。

如果在陈列公式之前有太多空白，有可能是因为你输入的时候在陈列公式之前留了一个空行。这个空行将开始一个新的段落，并且把 T<sub>E</sub>X 转入竖直模式。当 T<sub>E</sub>X 在竖直模式里遇到一个 '\$'，就会切换到水平模式并插入段间粘连 (`\parskip`)，接下来又插入一个行间粘连 (`\baselineskip`) `depth height`。然后当显示开始的时候，它插入粘连 (或者是 `\abovedisplayskip` 或者是 `\abovedisplayshortskip`, `depth height plus 0.1em minus 0.1em` 取决于前一行的长度)。只有最后一个粘

---

<sup>1</sup>

<sup>2</sup> T<sub>E</sub>X



连是你想要的。为了避免也得到段间粘连，在陈列公式之前不要留空行或者结束段落（比如用 `\par`）。

类似的，如果在陈列公式之后有太多的空白，可能是因为在输入时留了一个空行。去掉它即可。

如果使用宏得到的段落之后有太多竖直空白，可能里头包含了宏产生的段间粘连，一个空段，以及另外一个段间粘连。你可以通过在那个宏之后插入

```
\vskip -\parskip
\vskip -\baselineskip
```

来消除第二个段落间距。

有几个  $\text{T}_\text{E}\text{X}$  的参数——`\hangindent`，`\leftskip`，等等——影响  $\text{T}_\text{E}\text{X}$  塑段和折行的方式。Plain  $\text{T}_\text{E}\text{X}$  的命令，如 `\narrower` 和 `\hang`，会间接的用到这些参数；你也可以直接对它们赋值。如果你已经使用了这些命令中的一个（或者改变了这些参数中的一个），但是看上去对段落没有任何效果，问题可能在于，段落结束之前你先结束了一个组。例如：

```
{\narrower She very soon came to an open field, with
a wood on the other side of it: it looked much darker
than the last wood, and Alice felt a little timid
about going into it.}
```

这个段落不会变窄，因为最后那个右边大括号在  $\text{T}_\text{E}\text{X}$  拆段组行之前就结束了 `\narrower` 的组。作为补救，放一个 `\par` 在右边大括号之前；这样你才能看到想要的效果。

假设你有一些文字段落想放到页面的一个特定位置。显而易见的方式是用一个合适尺寸的水平盒子来封装段落，然后放到想要的位置。哎呀，这个办法不管用，因为  $\text{\TeX}$  不能在受限水平模式下断行。这么做，你将得到一个有些误导的错误消息，声称 `you're missing the end of a group`。摆脱这个限制的方法是使用：

```
\vbox{\hsize = <dimen> ... <paragraphs> ...}
```

其中， $\langle dimen \rangle$  是你希望的一行的长度。这么做就可以了，特别是当你想用盒子（带线框的盒子，不是  $\text{\TeX}$  盒子）封装一些段落的时候。

你可以使用 `\hrule` 和 `\vrule` 命令（第 [‘hrule’](#) 页）来画线。所需知道的是 (a) 在什么地方可以用哪个命令，以及 (b)  $\text{\TeX}$  如何在未明确指出的情况下决定线的长度。

- $\text{\TeX}$  处于竖直模式时，只能用 `\hrule`； $\text{\TeX}$  处于水平模式时，只能用 `\vrule`。所以你不能在水平盒子里使用横线或者在竖直盒子里使用竖线。然而，通过设定所有三个尺寸（dimension）`depth` `height`，你可以把横线画的又高又细，使它看起来像竖线。类似的，你也可以把竖线画的又矮又宽，使它看起来像横线。
- 如果你没有明确设定宽度，竖直盒子里的横线就会和盒子一样宽。水平盒子里的竖线也类似。如果你得到的线太长或太短，请检查外面盒子的尺寸。

一个例子，假设你想要

Help! Let  
me out of  
here!

The following input will do it:

```
\hbox{\vrule
```

```

\ vbox{\hrule \vskip 3pt
\ hbox{\hskip 3pt
\ vbox{\hsize = .7in \raggedright
\ noindent Help! Let me out of here!}%
\ hskip 3pt}%
\vskip 3pt \hrule}%
\vrule}

```

我们需要把文本放到竖直盒子里来让 T<sub>E</sub>X 把它当作段落处理。四层盒子是必要的---如果你有所怀疑，可以尝试在更少的盒子情况下运行这个例子。

你可以使用 `\headline` 和 `\footline` 命令（第 `'footline'` 页）来创建页眉和页脚，但是处理多于一行的页眉和页脚，就不大灵光。然而，通过重新定义 T<sub>E</sub>X 的输出例行程序的一些宏，就可以实现。

为了得到多行页眉，你需要做三件事：

- 1) 重新定义 `\makeheadline` 宏，T<sub>E</sub>X 的输出例行程序将会用到它。
- 2) 把额外的行所需的竖直距离加给 `\voffset`。
- 3) 对 `\vsize` 减少同样的大小。

下面的例子演示了你可以怎样做：

```

\advance\voffset by 2\baselineskip
\advance\vsize by -2\baselineskip
\def\makeheadline{\vbox to 0pt{\vss\ noindent
Header line 1\hfil Page \folio\break
Header line 2\hfil\break
Header line 3\hfil}%
\vskip\baselineskip}

```

一般你照着这个定义的样子做就可以了，只需要替换你自己的页眉内容并选一个合适的倍数给 `\baselineskip`（页眉行数减一）。

对多行页脚的处理也类似：

- 1) 重新定义 `\makefootline` 宏，T<sub>E</sub>X 的输出例行程序将会用到它。
- 2) 把额外的行所需的竖直距离加给 `\voffset`。
- 3) 从 `\vsize` 减去额外的行所需的竖直距离。

下面的例子演示了你可以怎样做：

```
\advance\vsiz by -2\baselineskip
\def\makefootline{%
 \lineskip = 24pt
 \vbox{\raggedright\noindent
 Footer line 1\hfil\break
 Footer line 2\hfil\break
 Footer line 3\hfil}}
```

同样，一般你照着这个定义的样子做就可以了。`\lineskip` 的值决定了页面上正文最后一行与页脚第一行的基行距离。

大多数情况下，如果有花括号匹配错误，`TEX` 会在你的输入文档里非常接近错误发生的地方给出诊断信息。但是，`TEX` 停止之前最令人沮丧的错误之一是

```
(\end occurred inside a group at level 1)
```

这表明在你的文档中有一个额外的左花括号或者缺少一个右花括号，但是并没有提示可能出现问题的地方。那么，怎么找到它呢？

一条有用的调试技巧是，在文档内部等距离地插入五到六处类似下面的内容（不要放在已知的编组里）：

```
}% a fake ending
```

假设问题是多了一个左花括号。如果它在第三和第四个伪装结尾（fake ending）之间，你将从前三个伪装结尾的地方得到出错消息，第四个则不会。原因是 `TEX` 将抱怨并忽略前三个伪装结尾，但是第四个正好匹配了那个额外的左花括号。这样你就知道了那个额外的左花括号在第三和第四个伪装结尾之间某个地方。如果嫌疑区域仍然太大，就去除已有的伪装结尾，并在嫌疑区域内重复上述步骤。如果问题是缺少一个右花括号，一旦你发现左边那个，就能确定右边的了。

这个方法不是必杀技。特别是，如果文档中有几个很大的组，这个方法就不那么灵了。然而大多数情况下，你可以适当地变化一下这个方法来找出暗地捣乱的花括号。

如果其他方法都不管用了，那就试着删掉文档后半部分（注意备份！）或者在中间插入一个 `\bye` 命令。如果问题仍然存在，那你就可以

确知它在前一半文档中；如果没有问题了，你也可以确知问题在后一般文档里。重复这个过程，最终你将会发现错误所在。

设置尺寸的最简单办法是直接赋值，比如：

```
\hsize = 6in
```

你也可以用其他尺寸来设定或者混用多种长度单位，这要多费一点事。有两种方法来创建这样的尺寸组合：

- 1) 你可以增加一个尺寸到另一个尺寸参数或寄存器。例如：

```
\hsize = 6in \advance\hsize by 3pc % 6in + 3pc
```

- 2) 你可以将一个尺寸取为另一个尺寸（或粘连）参数或寄存器的倍数。这时候， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  将丢掉粘连的伸长量和收缩量把它转换为尺寸。例如：

```
\parindent = .15\hsize
\advance\vsiz by -2\parskip
```

有时候新建一种“混合字体”会很有用处。混合字体命名为一个控制序列  $\mathcal{F}$ ，它的字符基于一种字体  $f_1$ ，但是有一些来自另一种字体  $f_2$ 。然后你可以像调用其他字体一样使用  $\mathcal{F}$  来设定文本使用混合字体。

要创建混合字体，你可以定义  $\mathcal{F}$  为一个宏。宏定义中，首先选择字体  $f_1$ ，然后定义控制序列来产生从  $f_2$  借来的字符。举个例子，假如你要创建一种混合字体 `\britrm`：基于 `cmr10` 但是其中的美元符号借用 `cmti10` 的英镑符号。英镑符号在 `cmti10` 的位置恰好跟美元符号在 `cmr10` 里的位置一样。你可以这样做：

```
\def\britrm{%
 \tenrm % \tenrm names the cmr10 font
```

```
\def\${\tenit\char ‘\$}}% \tenit names the cmti10 font.■
}
```

现在，当你使用 `\btrm` 字体的时候，`\$` 将生成英镑符号。

另一种方法会有同样的效果：首先改动类别码（category codes）来激活所考虑的字符，然后定义它们。例如：

```
\catcode ‘* = \active
\def*{\tentt \char ‘*}}
```

这个例子里，星号将从 `\tentt` 字体借用。如果接下来输入：

```
Debbie was the * of the show.
```

效果将会是

```
Debbie was the * of the show.
```

原文是“输入什么样，就是什么样”的文本。比较常见的是排版计算机输入，包括计算机程序和  $\text{T}_{\text{E}}\text{X}$  输入。计算机输入不易以原文呈现的原因是：

- 1) 有些字符（控制字符，转义符，花括号等）在  $\text{T}_{\text{E}}\text{X}$  里有特殊的含义。
- 2) 行尾和多个空格会被转换为一个空格。

为了呈现原文，你必须取消这些特殊含义和转换。最好的办法是使用宏。

要取消特殊含义，你可以改变那些字符的类别码。接下来的宏演示你如何做：

```
\chardef \other = 12
\def\deactivate{%
 \catcode‘\ = \other \catcode‘\{ = \other
 \catcode‘\} = \other \catcode‘\$ = \other
 \catcode‘\& = \other \catcode‘\# = \other
 \catcode‘\% = \other \catcode‘\~ = \other
 \catcode‘\^ = \other \catcode‘_ = \other
}
```

小心！一旦这样改变了类别码，就不能再使用任何控制序列了，因为现在不再有转义符了。你需要想法回到正常的操作模式。接下来，介绍完另一个问题：取消多个空格和行尾的转换以后，我们将予以解释。

Plain T<sub>E</sub>X 有两个命令：`\obeyspaces`（第 `\obeyspaces` 页）和 `\obeylines`（第 `\obeylines` 页）。几乎已经解决了上述问题，除了两件事：保留行首的空格以及保留多个空白行。对此，你需要更强大的方法——使用我们将要定义的 `\obeywhitespace` 宏。

T<sub>E</sub>X 总是收集多行文本为段落。为了原文呈现各行，一个办法是把每一行转换为独立的段落。<sup>3</sup> 你可以重新定义行尾字符来产生一个控制序列 `\par`。下面三个宏定义演示了如何做：

```
\def\makeactive#1{\catcode'#1 = \active \ignorespaces}
{% The group delimits the text over which ^^M is active.
 \makeactive\^^M %
 \gdef\obeywhitespace{%
 % Use \gdef so the definition survives the group.
 \makeactive\^^M %
 \let^^M = \newline %
 \aftergroup\removebox % Kill extra paragraph at end.
 \obeyspaces %
 }%
}
\def\newline{\par\indent}
\def\removebox{\setbox0=\lastbox}
```

一个要点是：`^^M` 必须被激活，无论是 `\obeywhitespace` 还是它的时候。

为了在原文文本之后回到正常的操作模式，你需要选定一个原文文本中几乎不用 的字符来作为临时的转义符。竖线 (`|`) `depth height plus 0.1em minus 0.1em` 经常是一个好的选择。为此，需要宏定义：

```
\def\verbatim{\par\begingroup\deactivate\obeywhitespace
 \catcode '\| = 0 % Make | the new escape character.
}

\def\endverbatim{\endgroup\endpar}

\def\||{|}
```

---

<sup>3</sup> `\break`

在原文文本中，你可以用双竖线 (`||`) `depth height plus 0.1em minus 0.1em` 表示单竖线，并用 `\endverbatim` 来结束原文部分。

这个技巧可以有多种变化：

- 如果一种编程语言有关键字，你可以把关键字变为一个命令，来产生粗体的关键字。这时候每一个关键字前面都要跟一个临时转义符。
- 如果有一个字符（假设是竖线）在原文文本中从来没有用到，你可以激活它并且用它结束原文部分。宏定义将类似于：

```
{\catcode '\| = \active
\gdef\verbatim{%
 \par\begingroup\deactivate\obeywhitespace
 \catcode '\| = \active
 \def |{\endgroup\par}%
}}
```

我们在这里只是提供了简单的办法来排版计算机程序。在反映程序的语法和语义环境方面，原文呈现经常不是一个清楚、易读的印刷习惯。如果你想在这个主题上走的更远，我们推荐阅读：

Baecker, Ronald M., and Marcus, Aaron, *Human Factors and Typography for More Readable Programs*. Reading, Mass.: Addison-Wesley, 1990.

如果 TeX 警告 “`forbidden control sequence`”，很有可能你在非外部环境中使用了外部宏（见 “`depth height`”，第 1 页）。一个外部宏的定义总是以 `\outer` 开始。外部宏不能被用于宏参量（argument）、宏定义、阵列的导言或者条件文本：只有当某个条件检验有特定的结果才是展开的文本。有些宏被以外部的方式定义是因为我们不想在这些环境里使用它们，一旦使用，必定出错。解决的办法就是重新定义宏或者移动它们到合适的环境里。

在不恰当的环境使用外部宏，还会引起 TeX 警告：不受控制（runaway）的情形或者不完备的条件。因为错误消息提供不了线索，这些问题通常难于诊断。如果你得到了这样的错误消息，找找看哪里用了



外部宏。可能你并不知道那些宏属于外部的，`\show\ a'`（第 `\show'` 页）■  
将显示 `\ a` 的定义，并告之 `\ a` 是否外部宏。

有时候需要在文档的部分内容里局部的改变类别码。例如，排版一个计算机程序或者将激活的字符用于专门目的。你需要反激活那些字符来使 `TEX` 可以像其他字符一样使用它们。

如果局部地改变了类别码，你可能会悲哀地发现 `TEX` 没有理会你的努力。通常，原因有两个：

- 1) `TEX` 决定输入字符的类别码，读到字符的时候就贴上相应的类别码。假设读到波浪符 (`~`) `depth height`，但是在 `TEX` 的胃还没有实际消化波浪符之前改变了它的类别码（见“`TEX`” `depth height`，第 `\texpouxi'` 页）。`TEX` 将会继续认定原来贴上的类别码。这样的问题通常是因为波浪符是一个宏定义的一部分，而那个宏里改变了波浪符的类别码。
- 2) 当 `TEX` 匹配一个宏引用和它的定义时，匹配的不仅仅是参数的字符而且包括他们的类别码。如果定义里字符的类别码与宏引用里的不同，`TEX` 不会认为字符匹配正确。这样的效果匪夷所思，因为是匹配的。例如，如果你已经定义了宏：

```
\def\eurodate#1/#2/#3{#2.#1.#3}
```

然后斜线字符在引用和定义 `\eurodate` 的时候必须有相同的类别码。

如果问题是因为嫌疑字符作为宏参量 (argument)，诊治方法是重新定义宏为一对宏 `\mstart` 和 `\mfinish`，分别在参量文本的前后应用。`\mstart` 设定类别码而 `\mfinish` 结束这一编组来取消变动。

为了使得宏文档易读，你可以在文档开头设定空格的类别码为 9（忽略字符），`\endlinechar`（第 `\endlinechar'` 页）为 -1。然后，你就可

以自由地使用空格和断行而不必担心产生多余的空白。忽略字符不会产生空白，但仍然决定控制序列定义。如果你真的需要一个空格，可以用 `\space` 命令（第 ``space'` 页）。

当然在文档最后，你需要恢复空格和行尾的类别码（分别是 10 和 5）`depth height`：可以把整个文档封装在一个编组里，也可以明确地恢复其正常值。如果使用编组封装，你还需要设定 `\globaldefs` 为 1 以使整个宏定义是全局的，在编组之外可见。

下面是这种格式宏文档的一个小例子：

```
\catcode '\ = 9 \endlinechar = -1

\def \makeblankbox #1 #2 {
 \hbox{\lower \dp0 \vbox{\hidehrule {#1} {#2}}
 \kern -#1 % overlap rules
 \hbox to \wd0{\hidevrule {#1} {#2}}%
 \raise \ht0 \vbox to #1{} % vrule height
 \lower \dp0 \vtop to #1{} % vrule depth
 \hfil \hidevrule {#2} {#1} }
 \kern -#1 \hidehrule {#2} {#1} } }

\def\hidehrule #1 #2 {
 \kern -#1 \hrule height#1 depth#2 \kern -#2 }
\def\hidevrule #1 #2 {
 \kern -#1 {\dimen0 = #1 \advance \dimen0 by #2
 \vrule width \dimen0 } \kern -#2 }

\catcode '\ = 10 \endlinechar = '\^^M
```

不改变类别码的话，这些宏需要写的更加紧凑，使用很少的空格和行尾更多的 ``%'`。







# 11



T<sub>E</sub>X T<sub>E</sub>X  
T<sub>E</sub>X T<sub>E</sub>X<sup>1</sup>

We skip \quid a little bit.

‘\quad’, 却输入成 ‘\quid’。此时 T<sub>E</sub>X 将给出下面的信息：

```
! Undefined control sequence.
1.291 We skip \quid
 a little bit.
?
```

此信息将同时出现在终端上和日志文件中。第一行始终以感叹号 (!) 开头，告诉你出现了什么问题。在 ‘?’ 提示符之前的最后两行（在此例子中同样也是接下来的两行）告诉你 T<sub>E</sub>X 在何处发现此错误。它在当前输入文件的第 291 行发现了错误，两个信息行之间的断行表示 T<sub>E</sub>X 在第 291 行的精确位置，即在 \quid 后面。当前输入文件是指，在运行时的终端输出上，最接近的非闭合左圆括号后显示的文件（见第 ‘infiles’ 页）。

这种错误，即控制序列未定义，是你最常出现的错误之一。如果你在提示符后键入另一个`?'，T<sub>E</sub>X 将显示下列信息：

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
```

这些选择各自的含义如下：

- 如果你键入 <return>，T<sub>E</sub>X 将继续处理你的文档。在此例子中它就忽略了 \quid。
- 如果你键入 `S'（或者 `s'——这时候大小写是等价的），T<sub>E</sub>X 将继续处理你的文档且不再暂停，找不到某文件。但错误信息还是会出现终端上和日志文件中。
- 如果你键入 `R' 或 `r'，你将得到与 `S' 相同的结果，但此时 T<sub>E</sub>X 即使找不到文件也不会暂停。
- 如果你键入 `Q' 或 `q'，T<sub>E</sub>X 将继续处理你的文档，且遇到错误后将不会再暂停，也不会再在终端上显示。错误信息还是会记录到日志文件中。
- 如果你键入 `X' 或 `x'，T<sub>E</sub>X 将尽力清理干净，丢弃正在构造的页面，并退出。你仍然可以打印或查看 T<sub>E</sub>X 已经生成的页面。
- 如果你键入 `E' 或 `e'，T<sub>E</sub>X 将如同 `X' 或 `x' 那样清理并中止，然后打开文本编辑器，跳转到错误所在的行。（并非所有系统都支持此选项。）
- 如果你键入 `H' 或 `h'，你将在终端上看到对此错误的进一步解释，可能还有对此错误的一些建议。这些信息同样会记录到你的日志文件中。对于上述例子的未定义控制序列，你将得到下列信息：

```
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
misspelled it (e.g., '\hobx'), type 'I' and the correct
spelling (e.g., 'I\hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
```

- 如果你键入 `?'，你将得到和上面相同的信息。

另外两个选择，即键入 ``I'` 或一个小整数，提供一种让 `TEX` 回到正常轨道上的方法，使得该错误不会在文档后面导致更多的错误：

- 如果你键入 ``I'` 或 ``i'` 并加上一些文本，`TEX` 将把文本插入错误位置之后，放在 `TEX` 处理的最内层级。在上面这个例子中，这表示在 `TEX` 的原始输入的位置，即在 ``\quid'` 之后。稍后我们将给出一个例子，说明把某些东西插入到最内层级与插入到原始输入的区别。在上面这个控制序列未定义的例子中，如果你键入：

```
I\quad
```

`TEX` 将执行 `\quad` 命令并在你需要的地方生成一个全方间隔。

- 如果你键入一个小于 100（错误信息让人误解为小于 10）的正整数，`TEX` 将从它处理的最内层级删除该数目的记号。（如果你键入大于或等于 100 的整数，`TEX` 将只删除 10 个记号！）

这里是另一种常见错误的例子：

```
Skip across \hskip 3cn by 3 centimeters.
```

这个例子的错误信息为：

```
! Illegal unit of measure (pt inserted).
<to be read again>
 c
<to be read again>
 n
1.340 Skip across \hskip 3cn
 by 3 centimeters.
```

这里 `TEX` 注意到 ``3'` 后面不是一个正确的度量单位，因此它假设该度量单位为点。`TEX` 将重新读取 ``cn'` 的记号并将它们插入你的输入中，而这并不是你想要的。在这种情形下若想得到更好的结果，你可以先键入 ``2'` 以跳过 ``cn'`。这样你将得到如下信息：

```
<recently read> n

1.340 Skip across \hskip 3cn
 by 3 centimeters.
```



现在你可以键入 `\hskip 3cm` 以得到你所要的间距（加上你已经得到的 3pt 间距）。<sup>2</sup>

如果你输入某些仅可用于数学模式的内容， $\text{\TeX}$  将帮你切换到数学模式，不管这是否你真正想要的。例如：

So `\spadesuit` s are trumps.

下面是  $\text{\TeX}$  的错误信息：

```
! Missing $ inserted.
<inserted text>
 $
<to be read again>
 \spadesuit
1.330 So \spadesuit
 s are trumps.
```

由于 `\spadesuit` 符号仅可在数学模式中使用， $\text{\TeX}$  已经在它前面插入一个 `\$`。在插入某个记号之后， $\text{\TeX}$  位于该记号的，在此情形中就是位于 `\$` 前面，并准备读取输入。键入 `\2` 将让  $\text{\TeX}$  跳过 `\$` 和 `\spadesuit` 记号，让它准备处理 `s are trumps.` 中的 `s`。（若你只是让  $\text{\TeX}$  继续，它将在数学模式中排版 `s are trumps`。）

在下面这个例子中， $\text{\TeX}$  对错误的诊断是完全不正确的：

`\hbox{One \vskip 1in two.}`

它给出的错误信息如下：

```
! Missing } inserted.
<inserted text>
 }
```

---

<sup>2</sup> `\I\unskip\hskip 3cm 3pt`

```

<to be read again>
 \vskip
1.29 \hbox{One \vskip
 1in two.}

```

问题在于，当  $\text{T}_{\text{E}}\text{X}$  位于受限水平模式中，即正在构建水平盒子时，不能使用 `\vskip`。但  $\text{T}_{\text{E}}\text{X}$  不是丢掉 `\vskip`，而是在它前面插入一个右花括号，以结束该水平盒子。如果你接受  $\text{T}_{\text{E}}\text{X}$  的修改， $\text{T}_{\text{E}}\text{X}$  ‘5’  
`’\vskip 1in’`

```

\leftline{Skip \smallskip a little further.} But no more.
\smallskip
! Missing } inserted.
<inserted text>
 }
<to be read again>
 \vskip
\smallskip ->\vskip
 \smallskipamount
<argument> Skip \smallskip
 a little further.
\leftline #1->\line {#1
 \hss }
1.93 ...Skip \smallskip a little further.}
 But no more.

```

plain  $\text{T}_{\text{E}}\text{X}$  \leftline  $\text{T}_{\text{E}}\text{X}$   $\text{T}_{\text{E}}\text{X}$  ‘!’

- 1)  $\text{T}_{\text{E}}\text{X}$
- 2)  $\text{T}_{\text{E}}\text{X}$  ‘\vskip’ \smallskip
- 3)  $\text{T}_{\text{E}}\text{X}$  \smallskip \smallskip \vskip ‘->’  $\text{T}_{\text{E}}\text{X}$
- 4)  $\text{T}_{\text{E}}\text{X}$  \smallskip  $\text{T}_{\text{E}}\text{X}$  \smallskip \leftline
- 5)  $\text{T}_{\text{E}}\text{X}$  \leftline  $\text{T}_{\text{E}}\text{X}$  #1 \leftline
- 6)  $\text{T}_{\text{E}}\text{X}$  ‘\vskip’  $\text{T}_{\text{E}}\text{X}$  \leftline \vskip \leftline

```
TeX TeX
\errorcontextlines 'errorcontextlines' TeX TeX \error-
contextlines 0
```

- <output> TeX
- <write> TeX \write TeX \write \shipout \write







# 12



eplain.tex plain T<sub>E</sub>X eplain.tex ‘resources’ eplain.tex

```
\def\makeactive#1{\catcode'#1 = \active \ignorespaces}%
\chardef\letter = 11 \chardef\other = 12
\def\uncatcodespecials{%
 \def\do##1{\catcode'##1 = \other}%
 \dospecials}% Defined in plain.
```

‘^^M’ \letreturn ‘^^M’ (它的定义在下面可以看到)。

```
{\makeactive\^^M \long\gdef\letreturn#1{\let^^M = #1}}%
```

这些宏用于吸收一个、两个或三个参量。

```
\def\gobble#1{}\def\gobbletwo#1#2{}%
\def\gobblethree#1#2#3{}%
```

现在我们建立一些惯例，用于读取这个文件的其他部分。在这个文件中我们允许名称带有 ‘@’ 的“私有”控制序列。这些控制序列在这个文件之外无法使用（除非你再次修改 ‘@’ 的类别码）。

```
\catcode'@ = \letter % Allow control sequences with @.
```

```

\let\@plainwlog = \wlog % Don't log register allocations.
\let\wlog = \gobble
\newlinechar = '^J

```

接下来两个宏提供了诊断输出的便捷方式。 `\loggingall` 打开所有追踪， 但让追踪输出仅写在日志文件中，而不在终端上显示。`\tracingboxes` 使得盒子被追踪时其内容完全显示。(T<sub>E</sub>X

```

\def\loggingall{\tracingcommands\tw@\tracingstats\tw@
\tracingpages\@ne\tracingoutput\@ne
\tracinglostchars\@ne\tracingmacros\tw@
\tracingparagraphs\@ne\tracingrestores\@ne
\showboxbreadth\maxdimen\showboxdepth\maxdimen}%
\def\tracingboxes{\showboxbreadth = \maxdimen
\showboxdepth = \maxdimen}%

```

0.4pt `\vruledefaultwidth\hruledefaultheight \hruledefaultdepth \hrule \vrule \ehrule \evrule` ‘e’ “eplain”。) 如果你显式给出尺寸 (比如 `\ehrule height 16pt`) `depth height` , T<sub>E</sub>X

```

\newdimen\hruledefaultheight \hruledefaultheight = 0.4pt
\newdimen\hruledefaultdepth \hruledefaultdepth = 0.0pt
\newdimen\vruledefaultwidth \vruledefaultwidth = 0.4pt
\def\ehrule{\hrule height\hruledefaultheight
depth\hruledefaultdepth}%
\def\evrule{\vrule width\vruledefaultwidth}%

\% '%' \write \percentchar ^^L

{\catcode'\% = \other \gdef\percentchar{%}}%
\def^^L{\par
}%

\tokstostring TEX \edef 'xrefs'
\@ttsA \@ttsB \spacesub

\def\tokstostring#1{\@ttsA#1 \ttsmarkA}%
\def\@ttsA#1 #2\ttsmarkA{\ifempty{#1}\else
\@ttsB #1\@ttsmarkB
\ifempty{#2}\else
\spacesub\@ttsA#2\ttsmarkA\fi\fi}%
\def\@ttsB#1{\ifx #1\@ttsmarkB\else
\string #1%

```

```

\expandafter\@ttsB\fi}%
\def\@ttsmarkB{\@ttsmarkB}% should never be expanded
\def\spacesub{+}%

\ifempty

\def\ifempty#1{\@ifempty #1\@emptymarkA\@emptymarkB}%
\def\@ifempty#1#2\@emptymarkB{\ifx #1\@emptymarkA}%
\def\@emptymarkA{\@emptymarkA}%

\for for 循环 TEX LATEX

\def\for#1:=#2\do#3{\edef\@fortmp{#2}%
 \ifx\@fortmp\empty \else
 \expandafter\@forloop#2,\@nil,\@nil\@@#1{#3}\fi}%
\def\@nnil{\@nil}%
\def\@fornoop#1\@@#2#3{}%
\def\@forloop#1,#2,#3\@@#4#5{\def#4{#1}\ifx #4\@nnil
 \else #5\def#4{#2} \ifx #4\@nnil \else
 #5\@forloop #3\@@#4{#5}\fi\fi}%
\def\@iforloop#1,#2\@@#3#4{\def#3{#1}\ifx #3\@nnil
 \let\@nextwhile=\@fornoop \else #4\relax
 \let\@nextwhile=\@iforloop\fi
 \@nextwhile#2\@@#3{#4}}%

\obeywhitespace \obeylines \obeyspaces

\def\alwayssspace{\hglue\fontdimen2\the\font \relax}%
{\makeactive\^^M \makeactive\ %
\gdef\obeywhitespace{%
\makeactive\^^M\def^^M{\par\indent}%
\aftergroup\@removebox% Kill extra paragraph at end.
\makeactive\ \let =\alwayssspace}}%
\def\@removebox{\setbox0=\lastbox}

\over “1/2” \frac The TEXbook 11.6 的答案。

\def\frac#1/#2{\leavevmode
 \kern.1em \raise .5ex \hbox{\the\scriptfont0 #1}%
 \kern-.1em $/$%
 \kern-.15em \lower .25ex \hbox{\the\scriptfont0 #2}}%

```

下面这些宏生成 T<sub>E</sub>X 圈中常用的标识。A<sub>M</sub>S-T<sub>E</sub>X 标识取自 *The T<sub>E</sub>Xbook* 第 420 页。L<sup>A</sup>T<sub>E</sub>X 标识取自 *latex.tex* 并稍作修改（我们修



改了 `A' 的字体); 类似地, BibTeX `\sevenrm METAFONT .mf`  
 METAFONT

Knuth, Donald E., *The METAFONTbook*. Reading, Mass.:  
 Addison-Wesley, 1986.

```
\def\LaTeX{L\kern-.26em \raise.6ex\hbox{\fiverm A}%
 \kern-.15em TeX}%
\def\AMSTeX{$_\cal A\kern-.1667em \lower.5ex\hbox{$_\cal M$}%
 \kern-.125em S$_-TeX}%
\def\BibTeX{{\rm B\kern-.05em {\sevenrm I\kern-.025em B}%
 \kern-.08em T\kern-.1667em \lower.7ex\hbox{E}%
 \kern-.125emX}}%
\font\mflogo = logo10
\def\MF{{\mflogo META}{\tenrm -}{\mflogo FONT}}%

\blackbox 'listmacs' \makeblankboxThe TeXbook 311

\def\blackbox{\vrule height .8ex width .6ex depth -.2ex}%
\def\makeblankbox#1#2{%
 \hbox{\lower\dp0\vbox{\hidehrule{#1}{#2}%
 \kern -#1% overlap rules
 \hbox to \wd0{\hidevrule{#1}{#2}%
 \raise\ht0\vbox to #1{}% vrule height
 \lower\dp0\vtop to #1{}% vrule depth
 \hfil\hidevrule{#2}{#1}}}%
 \kern-#1\hidehrule{#2}{#1}}}%
\def\hidehrule#1#2{\kern-#1\hrule height#1 depth#2
 \kern-#2}%
\def\hidevrule#1#2{\kern-#1{\dimen0 = #1
 \advance\dimen0 by #2 \vrule width\dimen0}\kern-#2}%

\numbername

\def\numbername#1{\ifcase#1%
 zero\or one\or two\or three\or four\or five%
 \or six\or seven\or eight\or nine\or ten\or #1\fi}%

\testfileexistence \jobname.#1 \iffileexists

\newif\iffileexists
\def\testfileexistence#1{\begingroup
 \immediate\openin0 = \jobname.#1\space
```

```

\ifeof 0\global\fileexistsfalse
\else \global\fileexiststrue\fi
\immediate\closein0
\endgroup}%

```

```

TeX $$ \leftdisplays \centereddisplays
\eqno \ifeqno \leqno \ifleqno \eqn \eq
The TeXbook 376

\newif\ifeqno \newif\ifleqno
\newtoks\@eqtoks \newtoks\@eqnotoks
\def\eq{\the\@eqtoks}\def\eqn{\the\@eqnotoks}%
\def\displaysetup#1$${%
 \@displaytest#1\eqno\eqno\@displaytest}%
\def\@displaytest#1\eqno#2\eqno#3\@displaytest{%
 \if #3% No \eqno, check for \leqno:
 \@displaytest#1\leqno\leqno\@displaytest
 \else
 \eqnotrue \leqnofalse % Have \eqno, not \leqno.
 \@eqnotoks = {#2}\@eqtoks = {#1}%
 \fi
 \generaldisplay$${}%
\def\@displaytest#1\leqno#2\leqno#3\@displaytest{%
 \@eqtoks = {#1}%
 \if #3%
 \eqnofalse % No \leqno; we're done.
 \else
 \eqnotrue \leqnotrue % Have \leqno.
 \@eqnotoks = {#2}%
 \fi}%

\leftdisplays \everydisplay \displaysetup \displaysetup
\generaldisplay

\newtoks\previousdisplay
\def\leftdisplays{%

```

```

\previouseverydisplay = \everydisplay
\everydisplay =
 {\the\previouseverydisplay \displaysetup}%
\def\generaldisplay{%
 \leftline{%
 \strut \indent \hskip\leftskip
 \dimen0 = \parindent
 \advance\dimen0 by \leftskip
 \advance\displaywidth by -\dimen0
 \@redefinealignmentdisplays
 \ifeqno \ifleqno
 \kern-\dimen0
 \rlap{$\displaystyle\eqn$}%
 \kern\dimen0
 \fi\fi
 $\displaystyle{\eq}$%
 \ifeqno \ifleqno\else
 \hfill $\displaystyle{\eqn}$%
 \fi\fi}}%
\def\centereddisplays{\let\displaysetup = \relax}%

\lefttdisplays \displaylines \eqalignno \leqalignno \eq
\halign \vcenter \halign \hfil plain.tex

\def\@redefinealignmentdisplays{%
 \def\displaylines##1{\displ@y
 \vcenter{\halign{\hbox to\displaywidth{$\@lign
 \displaystyle####\hfil$}\crrc##1\crrc}}}%
 \def\eqalignno##1{\displ@y
 \vcenter{\halign to\displaywidth{%
 $\@lign\displaystyle{####}$\tabskip\z@skip
 & $\@lign\displaystyle{{}####}$
 \hfil\tabskip\centering
 & \llap{$\@lign####$}\tabskip\z@skip\crrc
 ##1\crrc}}}%
 \def\leqalignno##1{\displ@y
 \vcenter{\halign to\displaywidth{%
 $\@lign\displaystyle{####}$\tabskip\z@skip
 & $\@lign\displaystyle{{}####}$

```

```

 $\hfil\tabskip\centering
&\kern-\displaywidth
\rlap{\kern-\parindent\kern-\leftskip$
\@lign####$}%
\tabskip\displaywidth\crr
##1\crr}}}%

```

TeX \time\day\month \year \monthname \timestring “1:14 p.m.”。 ■  
\timestamp 排印完整日期文本，类似于 “23 Apr 1964 1:14 p.m.”。

```

\def\monthname{%
 \ifcase\month
 \or Jan\or Feb\or Mar\or Apr\or May\or Jun%
 \or Jul\or Aug\or Sep\or Oct\or Nov\or Dec%
 \fi}%
\def\timestring{\begingroup
 \count0 = \time \divide\count0 by 60
 \count2 = \count0 % The hour.
 \count4 = \time \multiply\count0 by 60
 \advance\count4 by -\count0 % The minute.
 \ifnum\count4<10 \toks1 = {0}% Get a leading zero.
 \else
 \toks1 = {}%
 \fi
 \ifnum\count2<12 \toks0 = {a.m.}%
 \else
 \toks0 = {p.m.}%
 \advance\count2 by -12
 \fi
 \ifnum\count2=0 \count2 = 12 \fi % Make midnight ‘12’.
 \number\count2:\the\toks1 \number\count4
 \thinspace \the\toks0
\endgroup}%
\def\timestamp{\number\day\space\monthname\space
 \number\year\quad\timestring}%

```

`\numberedlist` 排印编号列表；`\endnumberedlist` 结束该列表。类似地，`\unorderedlist` 排印无序列表。这两种列表中的项目以 `\li` (“list item”) 开头。如果不需要列表项目间的额外间隔，你可以在列表开始处加上 `\listcompact`。列表可以任意嵌套。

更一般地，你可以通过改变下列寄存器的值控制列表项目间的间隔。如果列表项目经常比较长，你也许希望设定非零的 `\interitemskip`。各列表项目的左缩进量等于 `\parindent` 加 `\listleftindent`；各列表项目的右缩进量等于 `\listrightindent`。

```
\newskip\abovelistskip \abovelistskip = .5\baselineskip
\newskip\interitemskip \interitemskip = 0pt
\newskip\belowlistskip \belowlistskip = .5\baselineskip
\newdimen\listleftindent \listleftindent = \parindent
\newdimen\listrightindent \listrightindent = 0pt
\def\listcompact{\interitemskip = 0pt \relax}%
```

编号列表和无序列表都使用下面这些宏。我们并不改变 `\parindent` 的值，因为很多现有的宏比如 `\footnote` 依赖于 `\parindent`。我们必须考虑到项目包含不止一个段落的可能性。在这种情形中，除第一个之外的所有段落都将被缩进。我们用 `\leftskip` 和 `\rightskip` 缩进列表项目。我们通过改变 `\everydisplay` 设定陈列公式的缩进。

```
\newdimen\@listindent
\def\beginlist{%
 \@listindent = \parindent
 \advance\@listindent by \listleftindent
 \everydisplay = \expandafter{\the\everydisplay
 % Don't lose user's \everydisplay:
 \advance\displayindent by \@listindent
 \advance\displaywidth by -\@listindent
 \advance\displaywidth by -\listrightindent}%
 \nobreak\vskip\abovelistskip
 \parskip = 0pt
 % \leftskip shifts nested lists to the right on the page.
 \advance\leftskip by \@listindent
```

```

\advance\rightskip by \listrightindent}%
\def\printitem{\par\noindent
\llap{\hskip-\listleftindent \marker \enspace}}%
\def\endlist{\vskip\belowlistskip}%

```

通常重新定义 \numberedmarker 宏，你可以改变项目标签的样式。

```

\newcount\numberedlistdepth \newcount\itemnumber
\newcount\itemletter
\def\numberedmarker{%
\ifcase\numberedlistdepth
(impossible)%
\or \itemnumberout)%
\or \itemletterout)%
\else *%
\fi}%

```

这里是 \numberedlist 和 \unorderedlist 的定义。这两个定义的结构是相同的。

```

\def\numberedlist{\environment{@numbered-list}}%
\advance\numberedlistdepth by 1
\itemnumber = 1 \itemletter = 'a
\beginlist \let\marker = \numberedmarker
\def\li{%
\ifnum\itemnumber=1\else \vskip\interitemskip \fi
\printitem
\advance\itemnumber by 1 \advance\itemletter by 1
}%
\def\itemnumberout{\number\itemnumber}%
\def\itemletterout{\char\itemletter}%
\def\endnumberedlist{\par
\endenvironment{@numbered-list}\endlist}%

\newcount\unorderedlistdepth
\def\unorderedmarker{%
\ifcase\unorderedlistdepth
(impossible)%
\or \blackbox
\or ---%

```

```

\else *%
\fi}%
\def\unorderedlist{\environment{@unordered-list}%
\advance\unorderedlistdepth by 1
\beginlist \itemnumber = 1
\let\marker = \unorderedmarker
\def\li{%
\ifnum\itemnumber=1\else \vskip\interitemskip \fi
\printitem \advance\itemnumber by 1
}%
\def\endunorderedlist{\par
\endenvironment{@unordered-list}\endlist}%

```

`\listing` 宏用 `\tt` 字体排印指定文件的原文呈现。它基于 *The T<sub>E</sub>Xbook* 第 380 页中的代码。制表符生成固定大小的间隔，而换页符生成一个分页。其他控制字符生成该字体所在位置的字符，这通常不太有用。通过重新定义 `\setuplistinghook`，你可以在读入文件前针对特别字体和/

```

\def\listing#1{%
\par \begingroup \@setuplisting \setuplistinghook
\input #1 \endgroup}%
\let\setuplistinghook = \empty
\def\@setuplisting{%
\uncatcodespecials
\obeywhitespace \makeactive\‘ \makeactive\^^I
\def^^L{\vfill\eject}\tt}%
{\makeactive\‘ \gdef{\relax\lq}}% Defeat ligatures.
{\makeactive\^^I\gdef^^I{\hskip8\fontdimen2\tt \relax}}%

```

`\writetocentry \jobname.toc \writetocentry “chapter”`, 用于构成所调用的宏的名称。第二个参量是显示在目录项中的文本。`\writetocentry` 添加页码到宏调用中。例如:

```
\writetocentry{chapter}{Introduction}
```

将生成下列这行:

```
\tocchapterentry{Introduction}{2}
```

到 `.toc` 文件中, 表示 'Introduction' 从第 2 页开始。

利用 `\writenumberedtocentry` 你还可以提供第三个参数, 比如章编号。例如:

```
\writenumberedtocentry{chapter}{The second chapter}{2}
```

将写入这一行:

```
\tocchapterentry{The second chapter}{2}{14}
```

你也可以自己用 `\write` 命令写入 `\tocfile`。<sup>1</sup>

```
\newwrite\tocfile \newif\iftocfileopened
\def\opentocfile{\iftocfileopened\else
 \tocfileopenedtrue
 \immediate\openout\tocfile = \jobname.toc
\fi}%
\def\writetocentry#1#2{\ifrewritetocfile
 \opentocfile
 \write\tocfile{%
 \expandafter\noexpand \csname toc#1entry\endcsname
 {#2}{\folio}}}%
\fi\ignorespaces}%
\def\writenumberedtocentry#1#2#3{\ifrewritetocfile
 \opentocfile
 \write\tocfile{%
 \expandafter\noexpand \csname toc#1entry\endcsname
 {#2}{#3}{\folio}}}%
\fi\ignorespaces}%
```

要排印出目录, 只需用 `\readtocfile` 读取 `.toc` 文件。你应该在首次使用 `\writetocentry` 之前调用 `\readtocfile`。在处理目录且不想

---

<sup>1</sup> `\eplain 1.9 \writetocentry \writenumberedtocentry \writenumberedtocentry`  
`\tocfile \eplain`



重新生成它时，务必不要改写 .toc 文件——如果你这样做，文件内容将会丢失。命令 `\rewritetocfilefalse` 将禁止这种改写。

```
\newif\ifrewritetocfile \rewritetocfiletrue
\def\readtocfile{\testfileexistence{toc}%
 \iffileexists
 \input \jobname.toc
 \ifrewritetocfile \opentocfile \fi
 \fi}%
```

这里给出可能出现的 `\toc...entry` 宏的一些定义。这些定义只是作为例子而已——在目录中使用指引线通常不是最好的做法。

```
\def\tocchapterentry#1#2{\line{\bf #1 \dotfill\ #2}}%
\def\tocsectionentry#1#2{%
 \line{\quad\sl #1 \dotfill\ \rm #2}}%
\def\tocsubsectionentry#1#2{%
 \line{\qqquad\rm #1 \dotfill\ #2}}%
```

接下来的宏提供了符号式的交叉引用，让你可以在文档中通过名称而非实际页码提及其它部分的东西。`\xrdef{foo}` 定义标签 `foo` 为当前页码，而 `\xrefn{foo}` 生成该页码，比如 77。你更常用的是类似“see p. 77”的写法，因此 `\xref{foo}` 生成“p. 77”。如果 `foo` 未定义，就会得到一个警告信息。`\xrefwarningfalse` 取消这种警告。

这些宏没有提供对重复定义的保护。要检查重复定义，你可以将交叉引用文件排序，并用工具或肉眼检查同一个符号的相邻定义。

```
\newif\ifxrefwarning \xrefwarningtrue
\def\xrdef#1{\begingroup
 \xrlabel{#1}%
 \edef\@wr{\@writexrdef{\the\xrlabeltoks}}%
 \@wr
 \endgroup \ignorespaces}%
\def\@writexrdef#1{\write\reffile{%
 \string\gdef
```

```

\expandafter\string\csname#1\endcsname
{\noexpand\folio}\percentchar}}%
\def\xrefnumber#1{%
 \xrlabel{#1}%
 % \@xrlabeltoks now has the control sequence name.
 \toks0 =
 \expandafter{\csname\the\@xrlabeltoks\endcsname}%
 \expandafter \ifx\the\toks0\relax
 \ifxrefwarning \message{Undefined label
 '\tokstostring{#1}'.}\fi
 {\let\spacesub = \space
 \expandafter\xdef\the\toks0
 {\tt \tokstostring{#1}}}\fi
 \the\toks0}%
 \def\xref#1{p.\thinspace\xrefnumber{#1}}%
 \def\xrefn#1{\xrefnumber{#1}}%

```

这个宏将一个标签转换为一列字符记号，并放在寄存器 \labeltoks 中。除了普通字符之外，标签还可以包含空格和控制序列，但不能包含花括号。

```

\newtoks\@xrlabeltoks
\def\xrlabel#1{\begingroup
 \escapechar = '_ \edef\tts{\tokstostring{#1_}}%
 \global\@xrlabeltoks = \expandafter{\tts}%
 \endgroup}%

```

需要运行两遍才能得到正确的交叉引用，因为这些定义是写出到辅助文件 \jobname.aux 里的。 \readreffile 命令用于将它们读取回来。如果你不在第一个定义之前执行此命令，你将丢失上一遍运行得到的定义。

```

\newwrite\reffile \newif\ifreffileopened
\def\openreffile{\ifreffileopened\else
 \reffileopenedtrue
 \immediate\openout\reffile = \jobname.aux
 \fi}%
\def\readreffile{%
 \testfileexistence{aux}%
 \iffileexists
 \begingroup

```

```

 \@setletters
 \input \jobname.aux
 \endgroup
\else
 \message{No cross-reference file; I won't give you
 warnings about undefined labels.}%
 \xrefwarningfalse
\fi
\openreffile}%
\def\@setletters{%
 \catcode'_ = \letter \catcode'\+ = \letter
 \catcode'\- = \letter \catcode'\@ = \letter
 \catcode'\0 = \letter \catcode'\1 = \letter
 \catcode'\2 = \letter \catcode'\3 = \letter
 \catcode'\4 = \letter \catcode'\5 = \letter
 \catcode'\6 = \letter \catcode'\7 = \letter
 \catcode'\8 = \letter \catcode'\9 = \letter
 \catcode'\(= \letter \catcode'\) = \letter}%

```

按同样的方式，你可以用 `\eqdef` 和 `\eqref` 给出公式的符号式名称。`\eqdef` 插入自己的 `\eqno` 命令，因此它必须用在能用 `\eqno` 的地方。

```

\newcount\eqnumber
\def\eqdef#1{\global\advance\eqnumber by 1
 \expandafter\xdef
 \csname#1eqref\endcsname{\the\eqnumber}%
 \immediate\write\reffile{\string\def
 \expandafter\string\csname#1eqref\endcsname
 {\the\eqnumber}}}%
\eqno
\eqprint{\the\eqnumber}}%

```

`\eqref` 生成“(公式编号)”。通过重新定义 `\eqprint`，你可以实现更加复杂的格式。举个例子，你可以重新定义它，使得公式编号中包含章编号。

```

\def\eqref#1{%
 \expandafter \ifx \csname#1eqref\endcsname \relax
 \ifxrefwarning \message{Undefined equation label
 '#1'.}\fi

```

```

\expandafter\def\csname#1eqref\endcsname{00}%
\else \eqprint{\csname#1eqref\endcsname}%
\fi}%
\def\eqprint#1{(#1)}%

```

这些宏让你可以将手稿的一部分定义为命名编组（环境）。类似  $\text{T}_{\text{E}}\text{X}$  的编组，这些编组也可以嵌套，而且实际上它们的嵌套可以与  $\text{T}_{\text{E}}\text{X}$  编组的嵌套相互交织。如果开始的名称和结束的名称不匹配，你将得到一个错误信息。这些宏如此设计，使得当你得到这种错误信息时，你有机会方便地定位错误的来源。

用 `\environment{foo}` 开始一个环境，用 `\endenvironment{foo}` 结束该环境，其中 `foo` 是该环境的名称。我们的宏稍微改进了 *The  $\text{T}_{\text{E}}\text{X}$ book* 练习 5.7，添加对 `\begingroup` 和 `\endgroup` 是否配对的检查，并确保 `\environment` 和 `\endenvironment` 能够匹配。

```

\def\environment#1{\ifx\@groupname\undefined\else
 \errhelp = \@unnamedendgrouphelp
 \errmessage{'\@groupname' was not closed by
 \string\endenvironment}\fi
\def\@groupname{#1}%
\begingroup
 \let\@groupname = \undefined \ignorespaces}%
\def\endenvironment#1{\endgroup
 \def\@thearg{#1}%
 \ifx\@groupname\@thearg
 \else
 \ifx\@groupname\undefined
 \errhelp = \@isolatedendenvironmenthelp
 \errmessage{Isolated
 \string\endenvironment\space for '#1'}%
 \else
 \errhelp = \@mismatchedenvironmenthelp
 \errmessage{Environment '#1' ended,

```

```

 but '@groupname' started}%
 \endgroup % Probably a typo in the names.
\fi
\fi
\let\@groupname = \undefined \ignorespaces}%

```

你也可以给上述这些错误定义它们各自的帮助信息。

```

\newhelp\@unnamedendgrouphelp{%
 Most likely, you just forgot an^^J%
 \string\endenvironment.
 Maybe you should try inserting another^^J%
 \string\endgroup to recover.}%
\newhelp\@isolatedendenvironmenthelp{%
 You ended an environment X, but^^J%
 no \string\environment\space to start it
 is anywhere in sight.^^J%
 You might also be at an
 \string\endenvironment\space that would match^^J%
 a \string\begin group, i.e., you forgot an
 \string\endgroup.}%
\newhelp\@mismatchedenvironmenthelp{%
 You started an environment X, but^^J%
 you ended it with Y. Maybe you made a typo
 in one or the other^^J%
 of the names.}%

```

某些环境应当不允许出现在其他环境内部。我们称这些环境为“外部环境”。`\checkenv` 检测是否不存在当前有效的外部环境，若存在的话给出警告信息。要使用 `\checkenv`，你必须在每个外部环境开头处执行 `\environmenttrue` 命令。

```

\newif\ifenvironment
\def\checkenv{%
 \ifenvironment
 \errhelp = \@interwovenenvhelp
 \errmessage{Interwoven environments}%
 \endgroup
\fi}%
\newhelp\@interwovenenvhelp{%
 Perhaps you forgot to end the previous^^J%

```

```
environment? I'm finishing off the current group,^^J%
hoping that will fix it.}%
```

这三个宏 `\flushleft`、`\flushright` 和 `\center` 将后面各行文本以指定方式对齐。这种命令必须单独写在一行中。命令和文本应该包含在一个编组中——编组的结束就表示文本的结束。整个编组作为一个段落排版，各行视情况在一边或两边用空白填充。空行依原样显示。

```
\begingroup
\catcode '\^^M = \active
\globaldefs = 1 %
\def\flushleft{\beforejustify %
\aftergroup\@endflushleft %
\def^^M{\null\hfil\break}%
\def\@eateol^^M{\}\@eateol}%
\def\flushright{\beforejustify %
\aftergroup\@endflushright %
\def^^M{\break\null\hfil}%
\def\@eateol^^M{\hfil\null}\@eateol}%
\def\center {\beforejustify %
\aftergroup\@endcenter %
\def^^M{\hfil\break\null\hfil}%
\def\@eateol^^M{\hfil\null}\@eateol}%
\endgroup
```

在 `\flushleft`、`\flushright` 和 `\center` 的定义中，下述这些命令出现在 `\aftergroup` 的后面；它们在编组结束后被调用，以执行一些必需的清理工作。

```
\def\@endflushleft{\unpenalty
{\parfillskip = 0pt plus 1 fil\par}%
\ignorespaces}%
\def\@endflushright{%
% Remove the \hfil\null\break we just put on.
\unskip \setbox0=\lastbox \unpenalty
% We have fil glue at the left of the line;
```

```

% \parfillskip shouldn't affect that.
{\parfillskip = 0pt \par}\ignorespaces}%
\def\@endcenter{%
% Remove the \hfil\null\break we just put on.
\unskip \setbox0=\lastbox \unpenalty
% We have fil glue at the left of the line;
% \parfillskip must balance it.
{\parfillskip = 0pt plus 1fil \par}\ignorespaces}%
\def\beforejustify{%
\par\noindent
\catcode'\^M = \active
\checkenv \environmenttrue}%

```

`\makecolumns` 宏允许你给出表格的所有元素，而无需担心如何分栏。例如，在输入一个冗长的按字母排序的名单，并分多栏显示时，你通常不知道在何处结束一栏开始下一栏。另外，如果添加了另一项，各栏的划分将会改变。

`\makecolumns` 有两个（定界）参量：表格元素的总数以及表格的栏数。例如，`\makecolumns 37/3:` 指定一个三栏表格，其元素为接下来的 37 行。要调整表格在页面上的位置，你可以修改 `\parindent` 以确定表格左侧的空白，以及 `\hsize` 以确定从页面左边缘到表格右侧的距离。要允许在 `\valign` 之前分页，你可以修改 `\abovecolumnspenalty`。■

```

\newcount\abovecolumnspenalty
\abovecolumnspenalty = 10000
\newcount\@linestogo % Lines remaining to process.
\newcount\@linestogoincolumn % Lines remaining in column.
\newcount\@columndepth % Number of lines in a column.
\newdimen\@columnwidth % Width of each column.
\newtoks\crtok \crtok = {\cr}%
\def\makecolumns#1/#2: {\par \begingroup
\@columndepth = #1 \advance\@columndepth by #2
\advance\@columndepth by -1
\divide \@columndepth by #2

```

```

\@linestogoincolumn = \@columndepth \@linestogo = #1
\def\@endcolumnactions{%
 \ifnum \@linestogo<2
 \the\crtok \egroup \endgroup \par
 % End \valign and \makecolumns.
 \else
 \global\advance\@linestogo by -1
 \ifnum\@linestogoincolumn<2
 \global\@linestogoincolumn = \@columndepth
 \the\crtok
 \else &\global\advance\@linestogoincolumn by -1
 \fi
 \fi}%
\makeactive\^^M\letreturn\@endcolumnactions
\@columnwidth = \hsize
\advance\@columnwidth by -\parindent
\divide\@columnwidth by #2
\penalty\abovetopmargin
\noindent % It's not a paragraph (usually).
\valign\bgroup
 &\hbox to \@columnwidth{\strut ##\hfil}\cr
}% The next end-of-line starts everything going.

```

脚注通常用升高的数字作为参考符号。我们定义的 `\numberedfootnote` 宏就是这样的。它还重新定义了 `\vfootnote`，以支持比 plain TeX 更一般的脚注格式。尺寸寄存器 `\footnotemarkseparation` 控制脚注符号（比如数字）与脚注文本之间的间隔。`\everyfootnote` 记号在每个脚注之前插入。

Plain TeX 中的 `\footnote` 和 `\vfootnote` 定义保留在 `\@plainfootnote` 和 `\@plainvfootnote` 中，以备不时之需。

```

\newcount\footnotenum \newtoks\everyfootnote
\newdimen\footnotemarkseparation

```



```

\footnotemarkseparation = .5em
\let\@plainfootnote = \footnote
\let\@plainvfootnote = \vfootnote
\def\vfootnote#1{\insert\footins\bgroup
 \interlinepenalty\interfootnotelinepenalty
 \splittopskip\ht\strutbox \splitmaxdepth\dp\strutbox
 \floatingpenalty\@MM
 \leftskip\z@skip \rightskip\z@skip \spaceskip\z@skip
 \xspaceskip\z@skip
 \everypar = {}%
 \the\everyfootnote
 \indent\llap{#1\kern\footnotemarkseparation}\footstrut
 \futurelet\next\fo@t}%
\def\numberedfootnote{\global\advance\footnotenum by 1
 \@plainfootnote{$^{\number\footnotenum}$}}%

```

`\doublecolumns` 命令开始双栏输出，而 `\singlecolumn` 命令恢复单栏输出。两者可以在同个页面上相互切换。用 `\abovedoublecolumnskip` 和 `\belowdoublecolumnskip` 指定的粘连插入到双栏素材的前面和后面。

它们的实现方式来源于 *The T<sub>E</sub>Xbook* 第 417 页。

```

\newskip\abovedoublecolumnskip
\newskip\belowdoublecolumnskip
\abovedoublecolumnskip = \bigskipamount
\belowdoublecolumnskip = \bigskipamount
\newdimen\gutter \gutter = 2pc
\newdimen\doublecolumnhsize \doublecolumnhsize = \hsize
\newbox\@partialpage \newdimen\singlecolumnhsize
\newdimen\singlecolumnvsize \newtoks\previousoutput
\def\doublecolumns{\par % Don't start in horizontal mode.
 \previousoutput = \expandafter{\the\output}
 \advance\doublecolumnhsize by -\gutter

```

```

\divide\doublecolumnhsize by 2
\output = {\global\setbox\@partialpage =
 \vbox{\unvbox255\vskip\abovedoublecolumnskip}}%
\pagegoal = \pagetotal \break % Expands \output above.
\output = {\doublecolumnoutput}%
\singlecolumnhsize = \hsize
\singlecolumnvsize = \vsize
\hsize = \doublecolumnhsize \vsize = 2\vsize}%

```

`\@doublecolumnsplit` 宏执行实际的分栏。插入项被当作单栏素材；如果这不是你想要的，你需要修改输出例行程序。在 `\@doublecolumnsplit` 完成之后，`\box255` 将包含双栏素材。双栏素材前面是调用 `\doublecolumns` 前的单栏素材。`\box4` 将包含无法放入该页面的素材。

```

\def\@doublecolumnsplit{%
 \splittopskip = \topskip \splitmaxdepth = \maxdepth
 \dimen0 = \singlecolumnvsize
 \advance\dimen0 by -\ht\@partialpage
 \advance\dimen0 by -\ht\footins
 \advance\dimen0 by -\skip\footins
 \advance\dimen0 by -\ht\topins
 \begingroup
 \vbadness = 10000
 \global\setbox1=\vsplit255 to \dimen0 \wd1=\hsize
 \global\setbox3=\vsplit255 to \dimen0 \wd3=\hsize
 \endgroup
 \global\setbox4=\vbox{\unvbox255
 \penalty\outputpenalty}%
 \global\setbox255=\vbox{\unvbox\@partialpage
 \hbox to \singlecolumnhsize{\box1\hfil\box3}%
 \vfill}}%

```

`\doublecolumnoutput` 是真正的输出例行程序。我们调用 `\output` 执行实际的盒子送出工作。

```
\def\doublecolumnoutput{\@doublecolumnsplit
 \hsize = \singlecolumnhsize \vsize = \singlecolumnvsize
 \previousoutput \unvbox4}%

\singlecolumn 恢复单栏排版。它假定已调用了 \doublecolumns。

\def\singlecolumn{\par % Don't start in horizontal mode.
 \output = {\global\setbox1 =
 \vbox{\unvbox255\vskip\abovedoublecolumnskip}}%
 \pagegoal = \pagetotal \break \setbox255 = \box1
 {\singlecolumnvsize = \ht255
 \divide\singlecolumnvsize by 2
 \advance\singlecolumnvsize by +\ht\@partialpage
 \advance\singlecolumnvsize by +\ht\footins
 \advance\singlecolumnvsize by +\skip\footins
 \advance\singlecolumnvsize by +\ht\topins
 \@doublecolumnsplit}%
 \hsize = \singlecolumnhsize
 \vsize = \singlecolumnvsize
 \output = \expandafter{\the\previousoutput}%
 \unvbox255}%
```

现在我们必须撤销开始的修改（见第 `'eplainconv'` 页）。我们还给出版本标识，它可以在 `\fmtname` 和 `\fmtversion` 中得到。

```
\let\wlog = \@plainwlog \catcode'@ = \other
\def\fmtname{eplain}%
{\edef\plainversion{\fmtversion}%
 \xdef\fmtversion{1.0: 15 May 1990
 (and plain \plainversion)}%
}%
```







`\TeX` plain `\TeX` plain `\TeX` ‘a’ ‘6’ ‘‘

- `\TeX` ‘‘
- 
- `\TeX \tracingonline \TeX` ‘‘ \$\$
- $x x \quad \backslash alpha \quad \alpha \alpha$

- \*`\_` (第 ``@space'` 页)
- `\!` (第 ``@shriek'` 页)
- `\"` ö (第 ``@quote'` 页)
- # (第 ``@msharp'` 页, ‘@asharp’)
- `\#` # (第 ``@pound'` 页)
- \$ (第 ``@mathform'` 页)
- `\$` \$ (第 ``@bucks'` 页)
- \* % (第 ``@comments'` 页)
- `\%` % (第 ``@percent'` 页)
- & (第 ``@and'` 页)
- `\&` & (第 ``@and'` 页)
- ,  $p'$  (第 ``@prime'` 页)
- `\'` é (第 ``@prime'` 页)
- `\*` (第 ``@star'` 页)
- `\+` (第 ``@plus'` 页)

$\backslash,$  (第 ``@comma'` 页)  
 $*\backslash-$  (第 ``@minus'` 页)  
 $\backslash.$   $\dot{n}$  (第 ``@dot'` 页)  
 $*\backslash/$  (第 ``@slash'` 页)  
 $\backslash;$  (第 ``@semi'` 页)  
 $\backslash=$   $\bar{r}$  (第 ``@equal'` 页)  
 $*\backslash$  (第 ``@backslash'` 页)  
 $\backslash>$  (第 ``@greater'` 页)  
 $\backslash^$  (第 ``@hat'` 页)  
 $\backslash^$   $\hat{o}$  (第 ``@hat'` 页)  
 $\backslash^L$   $\backslashpar$  (第 ``@par'` 页)  
 $*\backslash^M$  (第 ``@newline'` 页)  
 $\backslash_$  (第 ``@underscore'` 页)  
 $\backslash_$   $-$  (第 ``@underscore'` 页)  
 $\backslash'$   $\grave{e}$  (第 ``@lquote'` 页)  
 $\backslash\{$  (第 ``@lbrace'` 页)  
 $\backslash\{$   $\{$  (第 ``@lbrace'` 页)  
 $\backslash|$   $\|$  (第 ``@bar'` 页)  
 $\backslash}$  (第 ``@rbrace'` 页)  
 $\backslash}$   $\}$  (第 ``@rbrace'` 页)  
 $\backslash\sim$  (第 ``@not'` 页)  
 $\backslash\sim$   $\tilde{a}$  (第 ``@not'` 页)  
 $\backslashaa$   $\text{\AA}$  (第 ``aa'` 页)  
 $\backslashAA$   $\text{\AA}$  (第 ``AA'` 页)  
 $*\backslashabove$  (第 ``above'` 页)  
 $*\backslashabovedisplayshortskip$   $\text{\TeX}$  0 pt plus 3 pt (第 ``abovedisplayshortskip'` 页)■  
 $*\backslashabovedisplayskip$   $\text{\TeX}$  12 pt plus 3 pt minus 9 pt (第 ``abovedisplayskip'` 页)■  
 $*\backslashabovewithdelims$  (第 ``abovewithdelims'` 页)  
 $*\backslashaccent$  (第 ``accent'` 页)  
 $\backslashactive$  13 (第 ``active'` 页)  
 $\backslashacute$   $\acute{x}$  (第 ``acute'` 页)  
 $*\backslashadjdemerits$  10000 (第 ``adjdemerits'` 页)  
 $*\backslashadvance$   $\backslashcount$  (第 ``advance'` 页)  
 $\backslashadvancepageno$   $\backslashpageno$  (第 ``advancepageno'` 页)  
 $\backslashae$   $\text{\ae}$  (第 ``ae'` 页)

`\AE`  $\mathbb{A}$  (第 `\AE` 页)  
`*\afterassignment` (第 `\afterassignment` 页)  
`*\aftergroup` (第 `\aftergroup` 页)  
`\aleph`  $\aleph$  (第 `\aleph` 页)  
`\allowbreak` `\penalty0` (第 `\hallowbreak` 页, `\vallowbreak`)  
`\alpha`  $\alpha$  (第 `\alpha` 页)  
`\amalg`  $\amalg$  (第 `\amalg` 页)  
`\angle`  $\angle$  (第 `\angle` 页)  
`\approx`  $\approx$  (第 `\approx` 页)  
`\arccos`  $\arccos$  (第 `\arccos` 页)  
`\arcsin`  $\arcsin$  (第 `\arcsin` 页)  
`\arctan`  $\arctan$  (第 `\arctan` 页)  
`\arg`  $\arg$  (第 `\arg` 页)  
`\arrowvert` (第 `\arrowvert` 页)  
`\Arrowvert` (第 `\Arrowvert` 页)  
`\ast`  $*$  (第 `\ast` 页)  
`\asymp`  $\asymp$  (第 `\asymp` 页)  
`*\atop` (第 `\atop` 页)  
`*\atopwithdelims` (第 `\atopwithdelims` 页)  
`\b`  $\underline{x}$  (第 `\b` 页)  
`\backslash`  $\backslash$  (第 `\backslash` 页)  
`*\badness` (第 `\badness` 页)  
`\bar`  $\bar{x}$  (第 `\bar` 页)  
`*\baselineskip` 12pt (第 `\baselineskip` 页)  
`*\batchmode` (第 `\batchmode` 页)  
`*\begingroup` `\endgroup` (第 `\begingroup` 页)  
`\beginsection` (第 `\@beginsection` 页)  
`*\belowdisplayshortskip`  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  7pt plus 0.3pt minus 4pt  
(第 `\belowdisplayshortskip` 页)  
`*\belowdisplayskip`  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  12pt plus 3pt minus 9pt (第 `\belowdisplayskip` 页)■  
`\beta`  $\beta$  (第 `\beta` 页)  
`\bf` `\tenbf``\fam=\bffam` (第 `\bf` 页)  
`\bffam` (第 `\bffam` 页)  
`\bgroup` (第 `\bgroup` 页)  
`\big` (第 `\big` 页)



`\Big` 11.5pt (第 ``Big'` 页)  
`\bigbreak` `\penalty-200` `\bigskipamount` (第 ``bigbreak'` 页)  
`\bigcap`  $\cap$  (第 ``bigcap'` 页)  
`\bigcirc`  $\bigcirc$  (第 ``bigcirc'` 页)  
`\bigcup`  $\bigcup$  (第 ``bigcup'` 页)  
`\bigg` 14.5pt (第 ``bigg'` 页)  
`\Bigg` 17.5pt (第 ``Bigg'` 页)  
`\biggl` `\bigg` (第 ``biggl'` 页)  
`\Biggl` `\Bigg` (第 ``Biggl'` 页)  
`\biggm` `\bigg` (第 ``biggm'` 页)  
`\Biggm` `\Bigg` (第 ``Biggm'` 页)  
`\biggr` `\bigg` (第 ``biggr'` 页)  
`\Biggr` `\Bigg` (第 ``Biggr'` 页)  
`\bigl` `\big` (第 ``bigl'` 页)  
`\Bigl` `\Big` (第 ``Bigl'` 页)  
`\bigm` `\big` (第 ``bigm'` 页)  
`\Bigm` `\Big` (第 ``Bigm'` 页)  
`\bigodot`  $\odot$  (第 ``bigodot'` 页)  
`\bigoplus`  $\oplus$  (第 ``bigoplus'` 页)  
`\bigotimes`  $\otimes$  (第 ``bigotimes'` 页)  
`\bigr` `\big` (第 ``bigr'` 页)  
`\Bigr` `\Big` (第 ``Bigr'` 页)  
`\bigskip` `\bigskipamount` (第 ``bigskip'` 页)  
`\bigskipamount` 12pt plus 4pt minus 4pt (第 ``bigskipamount'` 页)  
`\bigsqcup`  $\sqcup$  (第 ``bigsqcup'` 页)  
`\bigtriangledown`  $\nabla$  (第 ``bigtriangledown'` 页)  
`\bigtriangleup`  $\triangle$  (第 ``bigtriangleup'` 页)  
`\biguplus`  $\uplus$  (第 ``biguplus'` 页)  
`\bigvee`  $\vee$  (第 ``bigvee'` 页)  
`\bigwedge`  $\wedge$  (第 ``bigwedge'` 页)  
`*\binoppenalty` 700 (第 ``binoppenalty'` 页)  
`\bmod`  $n \bmod 2$  (第 ``bmod'` 页)  
`\bordermatrix` (第 ``bordermatrix'` 页)  
`\bot`  $\perp$  (第 ``bot'` 页)  
`*\botmark` (第 ``botmark'` 页)

`\bowtie`  $\bowtie$  (第 ``bowtie'` 页)  
`*\box` (第 ``box'` 页)  
`*\boxmaxdepth` `vbox` `\maxdimen` (第 ``boxmaxdepth'` 页)  
`\brace`  $\$n\brace k\$ \{^n_k\}$  (第 ``brace'` 页)  
`\bracevert` (第 ``bracevert'` 页)  
`\brack`  $\$n\brack k\$ \left[ ^n_k \right]$  (第 ``brack'` 页)  
`\break` `\penalty-10000` (第 ``hbreak'` 页, ``vbreak'`)  
`\breve`  $\breve{x}$  (第 ``breve'` 页)  
`*\brokenpenalty` `100` (第 ``brokenpenalty'` 页)  
`\buildrel` (第 ``buildrel'` 页)  
`\bullet`  $\bullet$  (第 ``bullet'` 页)  
`\bye` `\vfill` `\supereject` `\end` (第 ``@bye'` 页)  
`\c`  $\varsigma$  (第 ``c'` 页)  
`\cal`  $\mathcal{XYZ}$  (第 ``cal'` 页)  
`\cap`  $\cap$  (第 ``cap'` 页)  
`\cases`  $\{ \dots$  (第 ``cases'` 页)  
`*\catcode` (第 ``catcode'` 页)  
`\cdot`  $\cdot$  (第 ``cdot'` 页)  
`\cdotp`  $\cdot$  (第 ``cdotp'` 页)  
`\cdots`  $\dots$  (第 ``cdots'` 页)  
`\centerline` (第 ``centerline'` 页)  
`*\char` (第 ``char'` 页)  
`*\chardef` `0` `255` (第 ``chardef'` 页)  
`\check`  $\check{x}$  (第 ``check'` 页)  
`\chi`  $\chi$  (第 ``chi'` 页)  
`\choose`  $\$n\choose k\$ \binom{n}{k}$  (第 ``choose'` 页)  
`\circ`  $\circ$  (第 ``circ'` 页)  
`*\cleaders` (第 ``cleaders'` 页)  
`\cleartabs` (第 ``cleartabs'` 页)  
`*\closein` (第 ``closein'` 页)  
`*\closeout` (第 ``closeout'` 页)  
`*\clubpenalty` `150` (第 ``clubpenalty'` 页)  
`\clubsuit`  $\clubsuit$  (第 ``clubsuit'` 页)  
`\colon`  $:$  (第 ``colon'` 页)  
`\cong`  $\cong$  (第 ``cong'` 页)

`\coprod`  $\coprod$  (第 ``coprod'` 页)  
`*\copy` `\box` (第 ``copy'` 页)  
`\copyright` © (第 ``copyright'` 页)  
`\cos`  $\cos$  (第 ``cos'` 页)  
`\cosh`  $\cosh$  (第 ``cosh'` 页)  
`\cot`  $\cot$  (第 ``cot'` 页)  
`\coth`  $\coth$  (第 ``coth'` 页)  
`*\count` (第 ``count'` 页)  
`*\countdef` `\count` (第 ``countdef'` 页)  
`*\cr` (第 ``cr'` 页)  
`*\crrcr` `\cr` `\noalign` `\cr` (第 ``crrcr'` 页)  
`\csc`  $\csc$  (第 ``csc'` 页)  
`*\csname` `\endcsname` (第 ``csname'` 页)  
`\cup`  $\cup$  (第 ``cup'` 页)  
`\d`  $\mathfrak{d}$  (第 ``d'` 页)  
`\dag`  $\dagger$  (第 ``dag'` 页)  
`\dagger`  $\dagger$  (第 ``dagger'` 页)  
`\dashv`  $\dashv$  (第 ``dashv'` 页)  
`*\day` (第 ``day'` 页)  
`\ddag`  $\ddagger$  (第 ``ddag'` 页)  
`\ddagger`  $\ddagger$  (第 ``ddagger'` 页)  
`\ddot`  $\ddot{x}$  (第 ``ddot'` 页)  
`\ddots`  $\ddots$  (第 ``ddots'` 页)  
`*\deadcycles` `\shipout` `\output` (第 ``deadcycles'` 页)  
`*\def` (第 ``def'` 页)  
`*\defaultthyphenchar` (第 ``defaultthyphenchar'` 页)  
`*\defaultskewchar` (第 ``defaultskewchar'` 页)  
`\deg`  $\deg$  (第 ``deg'` 页)  
`*\delcode` (第 ``delcode'` 页)  
`*\delimiter` (第 ``delimiter'` 页)  
`*\delimiterfactor` 1000 1000, 其默认值为 901 (第 ``delimiterfactor'` 页)■  
`*\delimitershortfall` 5pt (第 ``delimitershortfall'` 页)  
`\delta`  $\delta$  (第 ``delta'` 页)  
`\Delta`  $\Delta$  (第 ``Delta'` 页)  
`\det`  $\det$  (第 ``det'` 页)

`\diamond`  $\diamond$  (第 ``diamond'` 页)  
`\diamondsuit`  $\diamond$  (第 ``diamondsuit'` 页)  
`\dim` `dim` (第 ``dim'` 页)  
`*\dimen` (第 ``dimen'` 页)  
`*\dimendef` `\dimen` (第 ``dimendef'` 页)  
`*\discretionary` (第 ``discretionary'` 页)  
`*\displayindent`  $\text{\TeX}$  (第 ``displayindent'` 页)  
`*\displaylimits` (第 ``displaylimits'` 页)  
`\displaylines` (第 ``displaylines'` 页)  
`*\displaystyle` (第 ``displaystyle'` 页)  
`*\displaywidowpenalty` 50 (第 ``displaywidowpenalty'` 页)  
`*\displaywidth`  $\text{\TeX}$  (第 ``displaywidth'` 页)  
`\div`  $\div$  (第 ``div'` 页)  
`*\divide` `\count` (第 ``divide'` 页)  
`\dot`  $\dot{x}$  (第 ``dot'` 页)  
`\doteq`  $\doteq$  (第 ``doteq'` 页)  
`\dotfill` (第 ``dotfill'` 页)  
`\dots`  $x_1, \dots, x_n$  (第 ``dots'` 页)  
`*\doublehyphendemerits` 10000 (第 ``doublehyphendemerits'` 页)  
`\downarrow`  $\downarrow$  (第 ``downarrow'` 页)  
`\Downarrow`  $\Downarrow$  (第 ``Downarrow'` 页)  
`\downbracefill`  $\text{hbox: } \underbrace{\hspace{1cm}}$  (第 ``downbracefill'` 页)  
`*\dp` (第 ``dp'` 页)  
`*\dump` (第 ``dump'` 页)  
`*\edef` (第 ``edef'` 页)  
`\egroup` (第 ``egroup'` 页)  
`\eject` (第 ``eject'` 页)  
`\ell`  $\ell$  (第 ``ell'` 页)  
`*\else` (第 ``@else'` 页)  
`*\emergencystretch` `\tolerance` (第 ``emergencystretch'` 页)  
`\empty` (第 ``empty'` 页)  
`\emptyset`  $\emptyset$  (第 ``emptyset'` 页)  
`*\end` `\output` (第 ``end'` 页)  
`*\endcsname` `\csname` (第 ``endcsname'` 页)  
`\endgraf` `\par` (第 ``endgraf'` 页)

`*\endgroup` `\begingroup` (第 `\endgroup` 页)  
`*\endinput` (第 `\endinput` 页)  
`\endinsert` (第 `\endinsert` 页)  
`\endline` `\cr` (第 `\endline` 页)  
`*\endlinechar` `TEX` `^^M` (第 `\endlinechar` 页)  
`\enskip`  $\frac{1}{2}\text{em}$  (第 `\enskip` 页)  
`\enspace`  $\frac{1}{2}\text{em}$  (第 `\enspace` 页)  
`\epsilon` (第 `\epsilon` 页)  
`\eqalign` (第 `\eqalign` 页)  
`\eqalignno` (第 `\eqalignno` 页)  
`*\eqno` (第 `\eqno` 页)  
`\equiv` (第 `\equiv` 页)  
`*\errhelp` `\errmessage` `TEX` (第 `\errhelp` 页)  
`*\errmessage` (第 `\errmessage` 页)  
`*\errorcontextlines` `TEX` `5` (第 `\errorcontextlines` 页)  
`*\errorstopmode` (第 `\errorstopmode` 页)  
`*\escapechar` `TEX` (第 `\escapechar` 页)  
`\eta` (第 `\eta` 页)  
`*\everycr` `\cr \cr \noalign \cr` `TEX` (第 `\everycr` 页)  
`*\everydisplay` `TEX` (第 `\everydisplay` 页)  
`*\everyhbox` `TEX` (第 `\everyhbox` 页)  
`*\everyjob` `TEX` (第 `\everyjob` 页)  
`*\everymath` `TEX` (第 `\everymath` 页)  
`*\everypar` `TEX` (第 `\everypar` 页)  
`*\everyvbox` `TEX` (第 `\everyvbox` 页)  
`*\exhyphenpenalty` `50` (第 `\exhyphenpenalty` 页)  
`\exists` (第 `\exists` 页)  
`\exp` `exp` (第 `\exp` 页)  
`*\expandafter` (第 `\expandafter` 页)  
`*\fam` `TEX` `7` (第 `\fam` 页)  
`*\fi` (第 `\fi` 页)  
`\filbreak` `\filbreak` (第 `\filbreak` 页)  
`*\finalhyphendemerits` `5000` (第 `\finalhyphendemerits` 页)  
`*\firstmark` (第 `\firstmark` 页)  
`\fivebf` `5 cmbx5` (第 `\fivebf` 页)

`\fivei` 5 cmmi5 (第 ``fivei'` 页)  
`\fiverm` 5 cmr5 (第 ``fiverm'` 页)  
`\fivesy` 5 cmsy5 (第 ``fivesy'` 页)  
`\flat`  $\flat$  (第 ``flat'` 页)  
`*\floatingpenalty` 0 (第 ``floatingpenalty'` 页)  
`\fmtname` (第 ``fmtname'` 页)  
`\fmtversion` (第 ``fmtversion'` 页)  
`\folio` `\pageno` (第 ``folio'` 页)  
`*\font` (第 ``font'` 页)  
`*\fontdimen` (第 ``fontdimen'` 页)  
`*\fontname` (第 ``fontname'` 页)  
`\footline` (第 ``footline'` 页)  
`\footnote` (第 ``footnote'` 页)  
`\forall`  $\forall$  (第 ``forall'` 页)  
`\frenchspacing` (第 ``frenchspacing'` 页)  
`\frown`  $\frown$  (第 ``frown'` 页)  
`*\futurelet` (第 ``futurelet'` 页)  
`\gamma`  $\gamma$  (第 ``gamma'` 页)  
`\Gamma`  $\Gamma$  (第 ``Gamma'` 页)  
`\gcd`  $\gcd$  (第 ``gcd'` 页)  
`*\gdef` `\global\def` (第 ``gdef'` 页)  
`\ge`  $\geq$  (第 ``ge'` 页)  
`\geq`  $\geq$  (第 ``geq'` 页)  
`\gets`  $\leftarrow$  (第 ``gets'` 页)  
`\gg`  $\gg$  (第 ``gg'` 页)  
`*\global` (第 ``global'` 页)  
`*\globaldefs` `\global` (第 ``globaldefs'` 页)  
`\goodbreak` `\penalty-500` (第 ``goodbreak'` 页)  
`\grave`  $\grave{\text{a}}$  (第 ``grave'` 页)  
`\H`  $\mathcal{H}$  (第 ``H'` 页)  
`*\halign` (第 ``halign'` 页)  
`\hang` `\parindent` (第 ``hang'` 页)  
`*\hangafter` (第 ``hangafter'` 页)  
`*\hangindent` (第 ``hangindent'` 页)  
`\hat`  $\hat{x}$  (第 ``hat'` 页)

`*\hbadness 1000` (第 `\hbadness` 页)  
`\hbar \hbar` (第 `\hbar` 页)  
`*\hbox hbox` (第 `\hbox` 页)  
`\headline` (第 `\headline` 页)  
`\heartsuit ♡` (第 `\heartsuit` 页)  
`*\hfil` (第 `\hfil` 页)  
`*\hfill \hfil` (第 `\hfill` 页)  
`*\hfilneg` (第 `\hfilneg` 页)  
`*\hfuzz hbox 0.1pt` (第 `\hfuzz` 页)  
`\hglue` (第 `\hglue` 页)  
`\hidewidth \hidewidth` (第 `\hidewidth` 页)  
`*\hoffset` (第 `\hoffset` 页)  
`*\holdinginserts` (第 `\holdinginserts` 页)  
`\hom hom` (第 `\hom` 页)  
`\hookleftarrow ↵` (第 `\hookleftarrow` 页)  
`\hookrightarrow ↶` (第 `\hookrightarrow` 页)  
`\hphantom` (第 `\hphantom` 页)  
`*\hrule` (第 `\hrule` 页)  
`\hrulefill` (第 `\hrulefill` 页)  
`*\hsize 6.5in` (第 `\hsize` 页)  
`*\hskip` (第 `\hskip` 页)  
`*\hss` (第 `\hss` 页)  
`*\ht` (第 `\ht` 页)  
`*\hyphenation` (第 `\hyphenation` 页)  
`*\hyphenchar` (第 `\hyphenchar` 页)  
`*\hyphenpenalty 50` (第 `\hyphenpenalty` 页)  
`\i 'i'` (第 `\i` 页)  
`\ialign \tabskip \everycr \halign` (第 `\ialign` 页)  
`*\if` (第 `\if` 页)  
`*\ifcase n n` (第 `\ifcase` 页)  
`*\ifcat` (第 `\ifcat` 页)  
`*\ifdim` (第 `\ifdim` 页)  
`*\ifeof` (第 `\ifeof` 页)  
`\iff ⇔` (第 `\iff` 页)  
`*\iffalse` (第 `\iffalse` 页)

`*\ifhbox`  $\hbox$  (第 `\@ifhbox` 页)  
`*\ifhmode`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  (第 `\@ifhmode` 页)  
`*\ifinner`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  (第 `\@ifinner` 页)  
`*\ifmmode`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  (第 `\@ifmmode` 页)  
`*\ifnum` (第 `\@ifnum` 页)  
`*\ifodd` (第 `\@ifodd` 页)  
`*\iftrue` (第 `\@iftrue` 页)  
`*\ifvbox`  $\vbox$  (第 `\@ifvbox` 页)  
`*\ifvmode`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  (第 `\@ifvmode` 页)  
`*\ifvoid` (第 `\@ifvoid` 页)  
`*\ifx` (第 `\@ifx` 页)  
`*\ignorespaces` (第 `\ignorespaces` 页)  
`\Im`  $\Im$  (第 `\Im` 页)  
`\imath`  $\imath$  (第 `\imath` 页)  
`*\immediate` (第 `\immediate` 页)  
`\in`  $\in$  (第 `\in` 页)  
`*\indent` `\parindent` (第 `\indent` 页)  
`\inf`  $\inf$  (第 `\inf` 页)  
`\infty`  $\infty$  (第 `\infty` 页)  
`*\input` (第 `\input` 页)  
`*\inputlineno` (第 `\inputlineno` 页)  
`*\insert` (第 `\insert` 页)  
`*\insertpenalties` (第 `\insertpenalties` 页)  
`\int`  $\int$  (第 `\int` 页)  
`*\interlinepenalty` 0 (第 `\interlinepenalty` 页)  
`\iota`  $\iota$  (第 `\iota` 页)  
`\it` `\tenit` $\mathrm{T}_\mathrm{E}\mathrm{X} $\mathrm{T}_\mathrm{E}\mathrm{X} $\mathrm{T}_\mathrm{E}\mathrm{X} (第 `\it` 页)  
`\item` `\parindent` (第 `\item` 页)  
`\itemitem` `\item` 2`\parindent` (第 `\itemitem` 页)  
`\itfam` (第 `\itfam` 页)  
`\j`  $\j$  (第 `\j` 页)  
`\jmath`  $\jmath$  (第 `\jmath` 页)  
`*\jobname`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  (第 `\jobname` 页)  
`\jot` (第 `\jot` 页)  
`\kappa`  $\kappa$  (第 `\kappa` 页)$$$



`\ker`  $\ker$  (第 ``ker'` 页)  
`*\kern` (第 ``kern'` 页)  
`\l`  $l$  (第 ``l'` 页)  
`\L`  $L$  (第 ``L'` 页)  
`\lambda`  $\lambda$  (第 ``lambda'` 页)  
`\Lambda`  $\Lambda$  (第 ``Lambda'` 页)  
`\land`  $\wedge$  (第 ``land'` 页)  
`\langle`  $\langle$  (第 ``langle'` 页)  
`*\language` (第 ``language'` 页)  
`*\lastbox` (第 ``lastbox'` 页)  
`*\lastkern` (第 ``lastkern'` 页)  
`*\lastpenalty` (第 ``lastpenalty'` 页)  
`*\lastskip` (第 ``lastskip'` 页)  
`\lbrace`  $\{$  (第 ``lbrace'` 页)  
`\lbrack`  $[$  (第 ``lbrack'` 页)  
`*\lccode` (第 ``lccode'` 页)  
`\lceil`  $\lceil$  (第 ``lceil'` 页)  
`\ldotp`  $\cdot$  (第 ``ldotp'` 页)  
`\ldots`  $\dots$  (第 ``ldots'` 页)  
`\le`  $\leq$  (第 ``le'` 页)  
`*\leaders` (第 ``leaders'` 页)  
`*\left` `\right` (第 ``left'` 页)  
`\leftarrow`  $\leftarrow$  (第 ``leftarrow'` 页)  
`\Leftarrow`  $\Leftarrow$  (第 ``Leftarrow'` 页)  
`\leftarrowfill` `\leftarrow hbox:`  $\longleftarrow$  (第 ``leftarrowfill'` 页)  
`\leftharpoondown`  $\leftharpoonup$  (第 ``leftharpoondown'` 页)  
`\leftharpoonup`  $\leftharpoonup$  (第 ``leftharpoonup'` 页)  
`*\lefthyphenmin`  $\TeX$  2 (第 ``lefthyphenmin'` 页)  
`\leftline` (第 ``leftline'` 页)  
`\leftrightarrow`  $\leftrightarrow$  (第 ``leftrightarrow'` 页)  
`\Leftrightarrow`  $\Leftrightarrow$  (第 ``Leftrightarrow'` 页)  
`*\leftskip`  $\TeX$  (第 ``leftskip'` 页)  
`\leq`  $\leq$  (第 ``leq'` 页)  
`\leqalignno` (第 ``leqalignno'` 页)  
`*\leqno` (第 ``leqno'` 页)

`*\let` (第 `\let` 页)  
`\lfloor`  $\lfloor$  (第 `\lfloor` 页)  
`\lg`  $\lg$  (第 `\lg` 页)  
`\lggroup`  $\left($  (第 `\lggroup` 页)  
`\lim`  $\lim$  (第 `\lim` 页)  
`\liminf`  $\liminf$  (第 `\liminf` 页)  
`*\limits` (第 `\limits` 页)  
`\limsup`  $\limsup$  (第 `\limsup` 页)  
`\line` (第 `\line` 页)  
`*\linepenalty` 10 (第 `\linepenalty` 页)  
`*\lineskip` `\lineskiplimit` 1pt (第 `\lineskip` 页)  
`*\lineskiplimit` `\lineskip` `\baselineskip` 0pt (第 `\lineskiplimit` 页)■  
`\ll`  $\ll$  (第 `\ll` 页)  
`\llap` (第 `\llap` 页)  
`\lmoustache`  $\int$  (第 `\lmoustache` 页)  
`\ln`  $\ln$  (第 `\ln` 页)  
`\lnot`  $\neg$  (第 `\lnot` 页)  
`\log`  $\log$  (第 `\log` 页)  
`*\long` `\par` (第 `\long` 页)  
`\longleftarrow`  $\longleftarrow$  (第 `\longleftarrow` 页)  
`\Longleftarrow`  $\Longleftarrow$  (第 `\Longleftarrow` 页)  
`\longlefttrightarrow`  $\longleftrightarrow$  (第 `\longlefttrightarrow` 页)  
`\Longlefttrightarrow`  $\Longleftrightarrow$  (第 `\Longlefttrightarrow` 页)  
`\longmapsto`  $\longmapsto$  (第 `\longmapsto` 页)  
`\longrightarrow`  $\longrightarrow$  (第 `\longrightarrow` 页)  
`\Longrightarrow`  $\Longrightarrow$  (第 `\Longrightarrow` 页)  
`\loop` `\repeat` (第 `\loop` 页)  
`*\looseness` (第 `\looseness` 页)  
`\lor`  $\vee$  (第 `\lor` 页)  
`*\lower` (第 `\lower` 页)  
`*\lowercase` (第 `\lowercase` 页)  
`\lq`  $\lq$  (第 `\lq` 页)  
`*\mag` 1000 (第 `\mag` 页)  
`\magnification` `\mag` (第 `\magnification` 页)

`\magstep`  $n$   $1000 \cdot 1.2^n$  (第 `\magstep` 页)  
`\magstephalf`  $1000 \cdot \sqrt{1.2}$  (第 `\magstephalf` 页)  
`\mapsto`  $\mapsto$  (第 `\mapsto` 页)  
`*\mark` (第 `\mark` 页)  
`*\mathaccent` (第 `\mathaccent` 页)  
`*\mathbin` (第 `\mathbin` 页)  
`*\mathchar` (第 `\mathchar` 页)  
`*\mathchardef`  $0 \ 2^{15} - 1$  (第 `\mathchardef` 页)  
`*\mathchoice` (第 `\mathchoice` 页)  
`*\mathclose` (第 `\mathclose` 页)  
`*\mathcode` (第 `\mathcode` 页)  
`*\mathinner` (第 `\mathinner` 页)  
`*\mathop` (第 `\mathop` 页)  
`*\mathopen` (第 `\mathopen` 页)  
`*\mathord` (第 `\mathord` 页)  
`\mathpalette` `\mathchoice` (第 `\mathpalette` 页)  
`*\mathpunct` (第 `\mathpunct` 页)  
`*\mathrel` (第 `\mathrel` 页)  
`\mathstrut` (第 `\mathstrut` 页)  
`*\mathsurround`  $\text{\TeX}$  (第 `\mathsurround` 页)  
`\matrix` (第 `\matrix` 页)  
`\max`  $\max$  (第 `\max` 页)  
`*\maxdeadcycles`  $\text{\TeX}$  `\deadcycles` 25 (第 `\maxdeadcycles` 页)  
`*\maxdepth` 4pt (第 `\maxdepth` 页)  
`\maxdimen`  $\text{\TeX}$  (第 `\maxdimen` 页)  
`*\meaning` (第 `\meaning` 页)  
`\medbreak` `\penalty-100` `\medskipamount` (第 `\medbreak` 页)  
`*\medmuskip` 4mu plus 2mu minus 4mu (第 `\medmuskip` 页)  
`\medskip` `\medskipamount` (第 `\medskip` 页)  
`\medskipamount` 6pt plus 2pt minus 2pt (第 `\medskipamount` 页)  
`*\message` (第 `\message` 页)  
`\mid`  $|$  (第 `\mid` 页)  
`\midinsert` (第 `\midinsert` 页)  
`\min`  $\min$  (第 `\min` 页)  
`\mit` `\fam=1` (第 `\mit` 页)

`*\mkern`  $\mu$  (第 ``mkern'` 页)  
`\models`  $\models$  (第 ``models'` 页)  
`*\month` (第 ``month'` 页)  
`*\moveleft` (第 ``moveleft'` 页)  
`*\moveright` (第 ``moveright'` 页)  
`\mp`  $\mp$  (第 ``mp'` 页)  
`*\mskip`  $\mu$  (第 ``mskip'` 页)  
`\mu`  $\mu$  (第 ``mu'` 页)  
`*\multiply` `\count` (第 ``multiply'` 页)  
`\multispan` (第 ``multispan'` 页)  
`*\muskip` (第 ``muskip'` 页)  
`*\muskipdef` `\muskip` (第 ``muskipdef'` 页)  
`\nabla`  $\nabla$  (第 ``nabla'` 页)  
`\narrower` `\parindent` (第 ``narrower'` 页)  
`\natural`  $\natural$  (第 ``natural'` 页)  
`\nearrow`  $\nearrow$  (第 ``nearrow'` 页)  
`\ne`  $\neq$  (第 ``ne'` 页)  
`\neg`  $\neg$  (第 ``neg'` 页)  
`\negthinspace`  $-\frac{1}{6}\text{em}$  (第 ``negthinspace'` 页)  
`\neq`  $\neq$  (第 ``neq'` 页)  
`\newbox` `\box` (第 ``@newbox'` 页)  
`\newcount` `\count` (第 ``@newcount'` 页)  
`\newdimen` `\dimen` (第 ``@newdimen'` 页)  
`\newfam` (第 ``@newfam'` 页)  
`\newhelp` (第 ``@newhelp'` 页)  
`\newif` (第 ``@newif'` 页)  
`\newinsert` `\box\count\dimen\skip` (第 ``@newinsert'` 页)  
`\newlanguage` `\language` (第 ``@newlanguage'` 页)  
`*\newlinechar` `\write` (第 ``newlinechar'` 页)  
`\newmuskip` `\muskip` (第 ``@newmuskip'` 页)  
`\newread` (第 ``@newread'` 页)  
`\newskip` `\skip` (第 ``@newskip'` 页)  
`\newtoks` `\toks` (第 ``@newtoks'` 页)  
`\newwrite` (第 ``@newwrite'` 页)  
`\ni`  $\ni$  (第 ``ni'` 页)

`*\noalign` (第 ``noalign'` 页)  
`*\noboundary` (第 ``noboundary'` 页)  
`\nobreak \penalty10000` (第 ``hnobreak'` 页, ``vnobreak'`)  
`*\noexpand` (第 ``noexpand'` 页)  
`*\noindent` (第 ``noindent'` 页)  
`\nointerlineskip` (第 ``nointerlineskip'` 页)  
`*\nolimits` (第 ``nolimits'` 页)  
`\nonfrenchspacing` (第 ``nonfrenchspacing'` 页)  
`*\nonscript` (第 ``nonscript'` 页)  
`*\nonstopmode` (第 ``nonstopmode'` 页)  
`\nopagenumbers \footline = \hfil` (第 ``nopagenumbers'` 页)  
`\normalbaselines \baselineskip\lineskip \lineskiplimit`  
(第 ``normalbaselines'` 页)  
`\normalbaselineskip \baselineskip` (第 ``normalbaselineskip'` 页)  
`\normalbottom` (第 ``normalbottom'` 页)  
`\normallineskip \lineskip` (第 ``normallineskip'` 页)  
`\normallineskiplimit \lineskiplimit` (第 ``normallineskiplimit'` 页)■  
`\not`  $\neq$  (第 ``not'` 页)  
`\notin`  $\notin$  (第 ``notin'` 页)  
`\nu`  $\nu$  (第 ``nu'` 页)  
`\null hbox` (第 ``null'` 页)  
`*\nulldelimiterspace 1.2pt` (第 ``nulldelimiterspace'` 页)  
`*\nullfont` (第 ``nullfont'` 页)  
`*\number` (第 ``number'` 页)  
`\nwarrow`  $\nwarrow$  (第 ``nwarrow'` 页)  
`\o`  $\emptyset$  (第 ``o'` 页)  
`\O`  $\emptyset$  (第 ``O'` 页)  
`\obeylines \par` (第 ``obeylines'` 页)  
`\obeyspaces` (第 ``obeyspaces'` 页)  
`\odot`  $\odot$  (第 ``odot'` 页)  
`\oe`  $\text{œ}$  (第 ``oe'` 页)  
`\OE`  $\text{Œ}$  (第 ``OE'` 页)  
`\offinterlineskip` (第 ``offinterlineskip'` 页)  
`\oint`  $\oint$  (第 ``oint'` 页)  
`\oldstyle 1234567890` (第 ``oldstyle'` 页)

`\omega`  $\omega$  (第 ``omega'` 页)  
`\Omega`  $\Omega$  (第 ``Omega'` 页)  
`\ominus`  $\ominus$  (第 ``ominus'` 页)  
`*\omit` (第 ``omit'` 页)  
`*\openin` (第 ``openin'` 页)  
`*\openout` (第 ``openout'` 页)  
`\openup` `\baselineskip` `\lineskip` `\lineskiplimit` (第 ``openup'` 页)  
`\oplus`  $\oplus$  (第 ``oplus'` 页)  
`*\or` `\ifcase` (第 ``@or'` 页)  
`\oslash`  $\oslash$  (第 ``oslash'` 页)  
`\otimes`  $\otimes$  (第 ``otimes'` 页)  
`*\outer` (第 ``outer'` 页)  
`*\output`  $\mathrm{T\!E\!X}$  (第 ``output'` 页)  
`*\outputpenalty` (第 ``outputpenalty'` 页)  
`*\over` (第 ``over'` 页)  
`\overbrace`  $\overbrace{h+w}$  (第 ``overbrace'` 页)  
`*\overfullrule` (第 ``overfullrule'` 页)  
`\overleftarrow`  $\overleftarrow{r+a}$  (第 ``overleftarrow'` 页)  
`*\overline`  $\overline{2b}$  (第 ``overline'` 页)  
`\overrightarrow`  $\overrightarrow{i+t}$  (第 ``overrightarrow'` 页)  
`*\overwithdelims` (第 ``overwithdelims'` 页)  
`\owns`  $\ni$  (第 ``owns'` 页)  
`\P`  $\P$  (第 ``P'` 页)  
`*\pagedepth`  $\mathrm{T\!E\!X}$  (第 ``pagedepth'` 页)  
`*\pagefilllstretch`  $\mathrm{T\!E\!X}$  `filll` (第 ``pagefilllstretch'` 页)  
`*\pagefillstretch`  $\mathrm{T\!E\!X}$  `fill` (第 ``pagefillstretch'` 页)  
`*\pagefilstretch`  $\mathrm{T\!E\!X}$  `fil` (第 ``pagefilstretch'` 页)  
`*\pagegoal`  $\mathrm{T\!E\!X}$  `\vsize` (第 ``pagegoal'` 页)  
`\pageinsert` (第 ``pageinsert'` 页)  
`\pageno` `\count0` (第 ``pageno'` 页)  
`*\pageshrink`  $\mathrm{T\!E\!X}$  (第 ``pageshrink'` 页)  
`*\pagestretch`  $\mathrm{T\!E\!X}$  (第 ``pagestretch'` 页)  
`*\pagetotal`  $\mathrm{T\!E\!X}$  (第 ``pagetotal'` 页)  
`*\par` (第 ``@par'` 页)  
`\parallel`  $\parallel$  (第 ``parallel'` 页)

`*\parfillskip` T<sub>E</sub>X (第 `\parfillskip` 页)  
`*\parindent` T<sub>E</sub>X (第 `\parindent` 页)  
`*\parshape` (第 `\parshape` 页)  
`*\parskip` T<sub>E</sub>X (第 `\parskip` 页)  
`\partial`  $\partial$  (第 `\partial` 页)  
`*\pausing` (第 `\pausing` 页)  
`*\penalty` (第 `\hpenalty` 页, `\vpenalty`)  
`\perp`  $\perp$  (第 `\perp` 页)  
`\phantom` (第 `\phantom` 页)  
`\phi`  $\phi$  (第 `\phi` 页)  
`\Phi`  $\Phi$  (第 `\Phi` 页)  
`\pi`  $\pi$  (第 `\pi` 页)  
`\Pi`  $\Pi$  (第 `\Pi` 页)  
`\plainoutput` plain T<sub>E</sub>X `\output` (第 `\plainoutput` 页)  
`\pm`  $\pm$  (第 `\pm` 页)  
`\pmatrix` (第 `\pmatrix` 页)  
`\pmod`  $x \equiv y + 1 \pmod{2}$  (第 `\pmod` 页)  
`*\postdisplaypenalty` 0 (第 `\postdisplaypenalty` 页)  
`\Pr` Pr (第 `\Pr` 页)  
`\prec`  $\prec$  (第 `\prec` 页)  
`\preceq`  $\preceq$  (第 `\preceq` 页)  
`*\predisplaypenalty` 0 (第 `\predisplaypenalty` 页)  
`*\preplaysize` T<sub>E</sub>X (第 `\preplaysize` 页)  
`*\pretolerance` 100 (第 `\pretolerance` 页)  
`*\prevdepth` (第 `\prevdepth` 页)  
`*\prevgraf` T<sub>E</sub>X (第 `\prevgraf` 页)  
`\prime`  $r'$  (第 `\prime` 页)  
`\proclaim` ... (第 `@proclaim` 页)  
`\prod`  $\prod$  (第 `\prod` 页)  
`\propto`  $\propto$  (第 `\propto` 页)  
`\psi`  $\psi$  (第 `\psi` 页)  
`\Psi`  $\Psi$  (第 `\Psi` 页)  
`\quad` 2em (第 `\quad` 页)  
`\quad` 1em (第 `\quad` 页)  
`*\radical` (第 `\radical` 页)

`\raggedbottom` (第 ``raggedbottom'` 页)  
`\raggedright` (第 ``raggedright'` 页)  
`*\raise` (第 ``raise'` 页)  
`\rangle` (第 ``rangle'` 页)  
`\rbrace` } (第 ``rbrace'` 页)  
`\rbrack` ] (第 ``rbrack'` 页)  
`\rceil` ] (第 ``rceil'` 页)  
`\Re`  $\Re$  (第 ``Re'` 页)  
`*\read` (第 ``read'` 页)  
`*\relax` (第 ``relax'` 页)  
`*\relpenalty` 500 (第 ``relpenalty'` 页)  
`\repeat` \loop (第 ``@repeat'` 页)  
`\rfloor` ] (第 ``rfloor'` 页)  
`\rgroup` ) (第 ``rgroup'` 页)  
`\rho`  $\rho$  (第 ``rho'` 页)  
`*\right` \left (第 ``right'` 页)  
`\rightarrow`  $\rightarrow$  (第 ``rightarrow'` 页)  
`\Rrightarrow`  $\Rightarrow$  (第 ``Rrightarrow'` 页)  
`\rightarrowfill` \rightarrow  $\longrightarrow$  (第 ``rightarrowfill'` 页)  
`\rightharpoondown`  $\searrow$  (第 ``rightharpoondown'` 页)  
`\rightharpoonup`  $\nearrow$  (第 ``rightharpoonup'` 页)  
`\rightleftharpoons`  $\rightleftharpoons$  (第 ``rightleftharpoons'` 页)  
`\rightline` (第 ``rightline'` 页)  
`*\rightskip`  $\TeX$  (第 ``rightskip'` 页)  
`*\righthyphenmin`  $\TeX$  3 (第 ``righthyphenmin'` 页)  
`\rlap` (第 ``rlap'` 页)  
`\rm` \tenrm \fam=0 (第 ``rm'` 页)  
`\rmoustache` ) (第 ``rmoustache'` 页)  
`\romannumeral` (第 ``romannumeral'` 页)  
`\root`  $\sqrt[3]{2}$  (第 ``root'` 页)  
`\rq` ' (第 ``rq'` 页)  
`\S` § (第 ``S'` 页)  
`\sb` (第 ``sb'` 页)  
`*\scriptfont` (第 ``scriptfont'` 页)




`*\scriptscriptfont` (第 ``scriptscriptfont'` 页)  
`*\scriptscriptstyle` (第 ``scriptscriptstyle'` 页)  
`*\scriptspace`  $\mathrm{T}_\mathrm{E}\mathrm{X}$  0.5pt (第 ``scriptspace'` 页)  
`*\scriptstyle` (第 ``scriptstyle'` 页)  
`*\scrollmode` (第 ``scrollmode'` 页)  
`\searrow` ↘ (第 ``searrow'` 页)  
`\sec` sec (第 ``sec'` 页)  
`*\setbox` (第 ``setbox'` 页)  
`*\setlanguage` `\language` (第 ``setlanguage'` 页)  
`\setminus` \ (第 ``setminus'` 页)  
`\settabs` (第 ``settabs'` 页)  
`\sevenbf` 7 cmbx7 (第 ``sevenbf'` 页)  
`\seveni` 7 cmmi7 (第 ``seveni'` 页)  
`\sevenrm` 7 cmr7 (第 ``sevenrm'` 页)  
`\sevensy` 7 cmsy7 (第 ``sevensy'` 页)  
`*\sfcode` (第 ``sfcode'` 页)  
`\sharp` ‡ (第 ``sharp'` 页)  
`*\shipout` .dvi (第 ``shipout'` 页)  
`*\show` (第 ``show'` 页)  
`*\showbox` (第 ``showbox'` 页)  
`*\showboxbreadth` 5 (第 ``showboxbreadth'` 页)  
`*\showboxdepth` 3 (第 ``showboxdepth'` 页)  
`\showhyphens` (第 ``showhyphens'` 页)  
`*\showlists` (第 ``showlists'` 页)  
`*\showthe` `\the` (第 ``showthe'` 页)  
`\sigma`  $\sigma$  (第 ``sigma'` 页)  
`\Sigma`  $\Sigma$  (第 ``Sigma'` 页)  
`\sim`  $\sim$  (第 ``sim'` 页)  
`\simeq`  $\simeq$  (第 ``simeq'` 页)  
`\sin` sin (第 ``sin'` 页)  
`\sinh` sinh (第 ``sinh'` 页)  
`\skew` (第 ``skew'` 页)  
`*\skewchar` (第 ``skewchar'` 页)  
`*\skip` (第 ``skip'` 页)  
`*\skipdef` `\skip` (第 ``skipdef'` 页)

`\sl`  $\backslash tensl \backslash fam = \backslash slfam$  (第 ``sl'` 页)  
`\slash`  $/$  (第 ``slash'` 页)  
`\slfam` (第 ``slfam'` 页)  
`\smallbreak`  $\backslash penalty - 50 \backslash smallskipamount$  (第 ``smallbreak'` 页)  
`\smallint`  $\int$  (第 ``smallint'` 页)  
`\smallskip`  $\backslash smallskipamount$  (第 ``smallskip'` 页)  
`\smallskipamount` 3 pt plus 1 pt minus 1 pt (第 ``smallskipamount'` 页)  
`\smash` (第 ``smash'` 页)  
`\smile`  $\smile$  (第 ``smile'` 页)  
`\sp` (第 ``sp'` 页)  
`\space` (第 ``space'` 页)  
`*\spacefactor` 1000 (第 ``spacefactor'` 页)  
`*\spaceskip`  $\backslash spacefactor < 2000$  (第 ``spaceskip'` 页)  
`\spadesuit`  $\spadesuit$  (第 ``spadesuit'` 页)  
`*\span` (第 ``span'` 页)  
`*\special` .dvi DVI (第 ``special'` 页)  
`*\splitbotmark`  $\backslash vsplit$  (第 ``splitbotmark'` 页)  
`*\splitfirstmark`  $\backslash vsplit$  (第 ``splitfirstmark'` 页)  
`*\splitmaxdepth`  $\backslash vsplit$  (第 ``splitmaxdepth'` 页)  
`*\splittopskip`  $\TeX \backslash vsplit$  (第 ``splittopskip'` 页)  
`\sqcap`  $\sqcap$  (第 ``sqcap'` 页)  
`\sqcup`  $\sqcup$  (第 ``sqcup'` 页)  
`\sqrt`  $\sqrt{2}$  (第 ``sqrt'` 页)  
`\sqsubseteq`  $\sqsubseteq$  (第 ``sqsubseteq'` 页)  
`\sqsupseteq`  $\sqsupseteq$  (第 ``sqsupseteq'` 页)  
`\ss`  $\beta$  (第 ``ss'` 页)  
`\star`  $\star$  (第 ``star'` 页)  
`*\string` (第 ``string'` 页)  
`\strut` (第 ``strut'` 页)  
`\subset`  $\subset$  (第 ``subset'` 页)  
`\subseteq`  $\subseteq$  (第 ``subseteq'` 页)  
`\succ`  $\succ$  (第 ``succ'` 页)  
`\succeq`  $\succeq$  (第 ``succeq'` 页)  
`\sum`  $\sum$  (第 ``sum'` 页)  
`\sup`  $\sup$  (第 ``sup'` 页)

`\supereject` (第 ``supereject'` 页)  
`\supset`  $\supset$  (第 ``supset'` 页)  
`\supseteq`  $\supseteq$  (第 ``supseteq'` 页)  
`\surd`  $\surd$  (第 ``surd'` 页)  
`\swarrow`  $\swarrow$  (第 ``swarrow'` 页)  
`\tau`  $\tau$  (第 ``t'` 页)  
`\tabalign`  $\backslash + \backslash outer$  (第 ``tabalign'` 页)  
`*\tabskip` (第 ``tabskip'` 页)  
`\tan`  $\tan$  (第 ``tan'` 页)  
`\tanh`  $\tanh$  (第 ``tanh'` 页)  
`\tau`  $\tau$  (第 ``tau'` 页)  
`\tenbf` 10 cmbx10 (第 ``tenbf'` 页)  
`\tenex` 10 cmex10 (第 ``tenex'` 页)  
`\teni` 10 cmmi10 (第 ``teni'` 页)  
`\tenit` 10 cmti10 (第 ``tenit'` 页)  
`\tenrm` 10 cmr10 (第 ``tenrm'` 页)  
`\tensl` 10 cmsl10 (第 ``tensl'` 页)  
`\tensy` 10 cmsy10 (第 ``tensy'` 页)  
`\tentt` 10 cmtt10 (第 ``tentt'` 页)  
`\TeX`  $\TeX$  (第 ``TeX'` 页)  
`*\textfont` (第 ``textfont'` 页)  
`\textindent`  $\backslash item$  (第 ``textindent'` 页)  
`*\textstyle` (第 ``textstyle'` 页)  
`*\the` (第 ``the'` 页)  
`\theta`  $\theta$  (第 ``theta'` 页)  
`\Theta`  $\Theta$  (第 ``Theta'` 页)  
`*\thickmuskip` 5 mu plus 5 mu (第 ``thickmuskip'` 页)  
`*\thinmuskip` 3 mu (第 ``thinmuskip'` 页)  
`\thinspace`  $\frac{1}{6}em$  (第 ``thinspace'` 页)  
`\tilde`  $\tilde$  (第 ``tilde'` 页)  
`*\time` (第 ``time'` 页)  
`\times`  $\times$  (第 ``times'` 页)  
`*\toks` (第 ``toks'` 页)  
`*\toksdef`  $\backslash toks$  (第 ``toksdef'` 页)  
`*\tolerance` (第 ``tolerance'` 页)

`\to` → (第 ``to'` 页)  
`\top` lattice top symbol:  $\top$  (第 ``top'` 页)  
`\topglue` (第 ``topglue'` 页)  
`\topinsert` (第 ``topinsert'` 页)  
`*\topmark` `\botmark` (第 ``topmark'` 页)  
`*\topskip` 10pt (第 ``topskip'` 页)  
`\tracingall` (第 ``tracingall'` 页)  
`*\tracingcommands` (第 ``tracingcommands'` 页)  
`*\tracinglostchars` (第 ``tracinglostchars'` 页)  
`*\tracingmacros` (第 ``tracingmacros'` 页)  
`*\tracingonline` (第 ``tracingonline'` 页)  
`*\tracingoutput` (第 ``tracingoutput'` 页)  
`*\tracingpages` (第 ``tracingpages'` 页)  
`*\tracingparagraphs` (第 ``tracingparagraphs'` 页)  
`*\tracingrestores` (第 ``tracingrestores'` 页)  
`*\tracingstats` (第 ``tracingstats'` 页)  
`\triangle`  $\triangle$  (第 ``triangle'` 页)  
`\triangleleft`  $\triangleleft$  (第 ``triangleleft'` 页)  
`\triangleright`  $\triangleright$  (第 ``triangleright'` 页)  
`\tt` `\tentt` $\text{\texttt{fam}}$  $=$  $\text{\texttt{ttfam}}$  (第 ``tt'` 页)  
`\ttfam` (第 ``ttfam'` 页)  
`\ttraggedright` (第 ``ttraggedright'` 页)  
`\u`  $\text{\text{r}}$  (第 ``u'` 页)  
`*\uccode` (第 ``uccode'` 页)  
`*\uchyph` (第 ``uchyph'` 页)  
`\underbar`  $\text{\text{fog}}$  (第 ``underbar'` 页)  
`\underbrace`  $\underbrace{x+x}$  (第 ``underbrace'` 页)  
`*\underline`  $\underline{x+y}$  (第 ``underline'` 页)  
`*\unhbox` (第 ``unhbox'` 页)  
`*\unhcopy` `\unhbox` (第 ``unhcopy'` 页)  
`*\unkern` (第 ``unkern'` 页)  
`*\unpenalty` (第 ``unpenalty'` 页)  
`*\unskip` (第 ``unskip'` 页)  
`*\unvbox` (第 ``unvbox'` 页)  
`*\unvcopy` `\unvbox` (第 ``unvcopy'` 页)

`\uparrow` ↑ (第 ``uparrow'` 页)  
`\Uparrow` ⇑ (第 ``Uparrow'` 页)  
`\upbracefill` hbox:  (第 ``upbracefill'` 页)  
`\updownarrow` ⇕ (第 ``updownarrow'` 页)  
`\Updownarrow` ⇕ (第 ``Updownarrow'` 页)  
`\uplus` ⊕ (第 ``uplus'` 页)  
`*\uppercase` (第 ``uppercase'` 页)  
`\upsilon` υ (第 ``upsilon'` 页)  
`\Upsilon` Υ (第 ``Upsilon'` 页)  
`\v` ö (第 ``v'` 页)  
`*\vadjust` (第 ``vadjust'` 页)  
`*\valign` (第 ``valign'` 页)  
`\varepsilon` ε (第 ``varepsilon'` 页)  
`\varphi` φ (第 ``varphi'` 页)  
`\varpi` ϖ (第 ``varpi'` 页)  
`\varrho` ϱ (第 ``varrho'` 页)  
`\varsigma` ς (第 ``varsigma'` 页)  
`\vartheta` ϑ (第 ``vartheta'` 页)  
`*\vbadness` 1000 (第 ``vbadness'` 页)  
`*\vbox` (第 ``vbox'` 页)  
`*\vcenter` (第 ``vcenter'` 页)  
`\vdash` ⊢ (第 ``vdash'` 页)  
`\vdots` ∴ (第 ``vdots'` 页)  
`\vec`  $\vec{x}$  (第 ``vec'` 页)  
`\vee` ∨ (第 ``vee'` 页)  
`\vert` | (第 ``vert'` 页)  
`\Vert` || (第 ``Vert'` 页)  
`*\vfil` (第 ``vfil'` 页)  
`*\vfill` `\vfil` (第 ``vfill'` 页)  
`*\vfilneg` (第 ``vfilneg'` 页)  
`\vfootnote` (第 ``vfootnote'` 页)  
`*\vfuzz` vbox 0.1pt (第 ``vfuzz'` 页)  
`\vglue` (第 ``vglue'` 页)  
`*\voffset` (第 ``voffset'` 页)  
`\vphantom` (第 ``vphantom'` 页)

`*\vrule` (第 `\vrule` 页)  
`*\vsize` 8.9in (第 `\vsize` 页)  
`*\vskip` (第 `\vskip` 页)  
`*\vsplit` (第 `\vsplit` 页)  
`*\vss` (第 `\vss` 页)  
`*\vtop` (第 `\vtop` 页)  
`*\wd` (第 `\wd` 页)  
`\wedge`  $\wedge$  (第 `\wedge` 页)  
`\widehat`  $y + \widehat{z} + a$  (第 `\widehat` 页)  
`\widetilde`  $b + \widetilde{c} + d$  (第 `\widetilde` 页)  
`*\widowpenalty` 150 (第 `\widowpenalty` 页)  
`\wlog` `\write` (第 `\wlog` 页)  
`\wp` Weierstraß ‘p’  $\wp$  (第 `\wp` 页)  
`\wr`  $\wr$  (第 `\wr` 页)  
`*\write` (第 `\write` 页)  
`*\xdef` `\global\edef` (第 `\xdef` 页)  
`\xi`  $\xi$  (第 `\xi` 页)  
`\Xi`  $\Xi$  (第 `\Xi` 页)  
`*\xleaders` (第 `\xleaders` 页)  
`*\xspaceskip` `\spacefactor`  $\geq 2000$  (第 `\xspaceskip` 页)  
`*\year` (第 `\year` 页)  
`\zeta`  $\zeta$  (第 `\zeta` 页)









# GNU

Version 1.3, 3 November 2008

Copyright © 2000,2001,2002,2007,2008 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program

should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, L<sup>A</sup>T<sub>E</sub>X input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”,

“Dedications”, “Endorsements”, or “History”). To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section ‘[copying](#)’.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may

add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the

Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the



Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting

from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

”Massive Multiauthor Collaboration Site” (or ”MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ”Massive Multiauthor Collaboration” (or ”MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

”CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

”Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ”eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## 11. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled ”GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ”with...Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being  
LIST.

If you have Invariant Sections without Cover Texts, or some  
other combination of the three, merge those two alternatives to suit  
the situation.

If your document contains nontrivial examples of program code, we  
recommend releasing these examples in parallel under your choice of free  
software license, such as the GNU General Public License, to permit  
their use in free software.



|





Paul W. Abrahams, Sc.D., CCP, is a consulting computer scientist and a past president of the Association for Computing Machinery. His specialties are programming languages, software systems design and implementation, and technical writing. He received his doctorate from the Massachusetts Institute of Technology in 1963 in mathematics, studying artificial intelligence under Marvin Minsky and John McCarthy. He is one of the designers of the first LISP system and the designer of the CIMS PL/I system, developed when he was a professor at New York University. More recently, he has designed SPLASH, a Systems Programming LAnguage for Software Hackers. Paul resides in Deerfield, Massachusetts, where he writes, hacks, hikes, hunts wild mushrooms, and listens to classical music.

Kathryn A. Hargreaves received her M.S. degree in computer science from the University of Massachusetts, Boston, in August 1989. Her specialties are digital typography and human vision. She developed a set of programs to produce high-quality, freely distributable digital type for the Free Software Foundation and also worked with Robert A. Morris as an Adjunct Research Associate. In 1986 she completed the Reentry Program in Computer Science for Women and Minorities at the University of California at Berkeley, where she also worked in the T<sub>E</sub>X research group under Michael Harrison. She has studied letterform design with Don Adleta, André Gürtler, and Christian Mengelt at the Rhode Island School of Design. A journeyman typographer, she has worked at Headliners/Identicolor, San Francisco, and Future Studio, Los Angeles, two leading typographical firms. She also holds an M.F.A. in Painting/Sculpture/Graphic Arts from the University of California at Los Angeles. Kathy paints watercolors, designs letterforms, plays piano, and reads feminist film criticism.

Like Kathy, Karl Berry received his M.S. degree in computer science from the University of Massachusetts, Boston, in August 1989. He also worked for the Free Software Foundation, did research with Morris, and has studied with Adleta, Gürtler, and Mengelt. He has been working with T<sub>E</sub>X since 1983 and has installed and maintained the

T<sub>E</sub>X system at a number of universities. He was the maintainer of the Web2c system developed by Tim Morgan for a number of years, among other T<sub>E</sub>X projects. He became the president of the T<sub>E</sub>X Users Group in 2003.

This book was composed using T<sub>E</sub>X (of course), developed by Donald E. Knuth. The main text is set in Computer Modern, also designed by Knuth. The heads of the original book were set in Zapf Humanist (the Bitstream version of Optima), designed by Hermann Zapf.

The paper was Amherst Ultra Matte 45 lb. The printing and binding were done by Arcadia Graphics-Halliday. The phototypeset output was produced at Type 2000, Inc., in Mill Valley, California. Proofs were made on an Apple LaserWriter Plus and on a Hewlett Packard LaserJet II.

Cross-referencing, indexing, and the table of contents were done mechanically, using the macros of ‘eplain’ together with additional macros custom-written for this book. The production of the index was supported by an additional program written in Icon.



|                     |    |                                      |
|---------------------|----|--------------------------------------|
| 46                  | 65 | 79                                   |
| 46                  | 66 | 79                                   |
| T <sub>E</sub> X 48 | 66 | 80                                   |
| 49                  | 66 | 80                                   |
| ASCII 50            | 66 | 80                                   |
| 50                  | 66 | 81                                   |
| 51                  | 67 | 81                                   |
| 51                  | 67 | plain T <sub>E</sub> X 81            |
| 51                  | 67 | 82                                   |
| 52                  | 67 | 82                                   |
| 54                  | 67 | 82                                   |
| 55                  | 67 | 82                                   |
| 55                  | 68 | 83                                   |
| 55                  | 68 | 83                                   |
| 56                  | 70 | 83                                   |
| 56                  | 70 | 83                                   |
| 56                  | 71 | 83                                   |
| 56                  | 71 | 84                                   |
| 56                  | 71 | 84                                   |
| 58                  | 74 | 85                                   |
| 58                  | 74 | T <sub>E</sub> X M <sub>E</sub> X 85 |
| 58                  | 75 | 85                                   |
| 59                  | 76 | 85                                   |
| 59                  | 76 | 85                                   |
| 60                  | 76 | 85                                   |
| 60                  | 76 | 86                                   |
| 61                  | 77 | 86                                   |
| 61                  | 77 | 86                                   |
| 62                  | 78 | 86                                   |
| 62                  | 78 | 87                                   |
| 63                  | 78 |                                      |
| 63                  | 79 |                                      |