

# siunitx —— 完整的国际单位制 (si) 宏包<sup>\*</sup>

Joseph Wright (约瑟夫·莱特)<sup>†</sup> 【著】

赣医一附院神经内科 黄旭华<sup>‡</sup> 【译】

分发于 2022-12-05

## 摘要

一个物理量 (physical quantities) 既有数字 (numbers) 又有单位 (units), 每个物理量都应该表示为一个数字和一个单位的乘积 (product)。排版物理量需要小心, 以确保数字-单位的组合 (number-unit combination) 的数学意义 (mathematical meaning) 是明确的。特别是 **si**<sup>1</sup> 系统规定了一组一致的单位 (units), 并规定了如何使用这些单位。然而, 不同的国家和出版商对数字 (和单位) 的确切外观 (exact appearance) 有不同的约定。**siunitx** 宏包为作者提供了一套工具, 以便以一致的方式排版数量。该宏包有一个扩展 (extended) 的配置选项 (configuration options) 集, 使其可以遵循不同的排版约定 (typographic conventions) 与相同的输入语法 (input syntax)。该宏包包括数字和单位的自动处理, 以及控制表格中数字对齐的能力。

---

<sup>\*</sup>本文件描述了 v3.1.11, 最新版修订于 2022-12-05。

<sup>†</sup>E-mail: [joseph.wright@morningstar2.co.uk](mailto:joseph.wright@morningstar2.co.uk)

<sup>‡</sup>一名业余  $\text{\LaTeX}$  爱好者。

<sup>1</sup>si: 法语 *Système International d'Unités* 的缩写, 国际单位制。

# 目 录

摘要	1
1 介绍	5
2 为急不可耐的人准备的 <b>siunitx</b>	6
3 使用 <b>siunitx</b> 宏包	8
3.1 数字	8
3.2 角度	9
3.3 单位	10
3.4 复数和数量	12
3.5 单位的宏	12
3.6 单位的缩写	17
3.7 创建新的宏	22
3.8 表格的材料	24
4 宏包的控制选项	28
4.1 键-值控制系统	28
4.2 打印	28
4.3 解析数字	32
4.4 数字后处理	36
4.5 打印数字	40
4.6 列表、乘积和范围	47
4.7 复数	51
4.8 角度	53
4.9 创建单位	56
4.10 使用单位	57
4.11 数量	63
4.12 表格的材料	65

4.13	局部设置选项	79
4.14	仅在前言中使用的选项	79
5	从版本 2 升级	80
6	BIPM 进行的单位变更	82
7	本地化	84
8	与其他宏包的兼容性	85
9	使用 <b>siunitx</b> 时的注意事项	86
9.1	有问题的字体编码	86
9.2	调整 <code>\litre</code> 和 <code>\liter</code>	86
9.3	确保文本或数学输出	86
9.4	在 <code>\text</code> 中包含文字连字符	87
9.5	展开表格中的内容	87
9.6	将 <b>siunitx</b> 与 <b>datatool</b> 一起使用	89
9.7	在节标题和书签中使用单位	91
9.8	在标题下视觉居中的左对齐列	91
9.9	回归表	92
9.10	最大化性能	94
9.11	对 <code>\kWh</code> 单位的特殊考虑	94
9.12	创建包含数字和单位的列	95
9.13	带有标题行的表格	97
9.14	将区域设置与 <b>babel</b> 语言关联	98
9.15	符号“数字 (digits)”	98
9.16	演示词头	98
9.17	创建一组预定义的单位	99
10	使用 (SI) 单位	100
10.1	单位	100
10.2	数学意义	101

10.3 图形和表格 . . . . .	103
11 安装	107
12 致谢	107
13 提出建议并报告错误	107
索引	109

# 1 介绍

计量单位 (units of measurement) 的正确应用在技术应用中非常重要。出于这个原因，国际计量大会 (*Conférence Générale des Poids et Mesures*, cgpm) 对一贯单位系统 (coherent units system) 进行了精心定义 (carefully-crafted definitions)，因此产生了国际单位制 (*Système International d'Unités*, si)。同时，为了确保印刷品 (printed matter) 不会失去意义，已经存在正确显示数字和单位的排版约定 (typographic conventions)。

siunitx 宏包旨在为 L<sup>A</sup>T<sub>E</sub>X 用户提供一种统一的方法，以便正确、轻松地设置数字 (numbers) 和单位 (units)。siunitx 宏包的设计理念是遵循默认的约定规则 (agreed rules)，但允许通过选项设置 (option settings) 进行更改。通过这种方式，用户可以使用 siunitx 来遵循出版商 (publishers)、合著者 (co-authors)、大学 (universities) 等的要求，而无需更改输入 (input)。

## 2 为急不可耐的人准备的 siunitx

该宏包提供了以下用户宏 (user macros):

- `\ang[⟨options⟩]{⟨angle⟩}`
- `\num[⟨options⟩]{⟨number⟩}`
- `\unit[⟨options⟩]{⟨unit⟩}`
- `\qty[⟨options⟩]{⟨number⟩}{⟨unit⟩}`
- `\numlist[⟨options⟩]{⟨numbers⟩}`
- `\numproduct[⟨options⟩]{⟨numbers⟩}`
- `\numrange[⟨options⟩]{⟨numbers⟩}{⟨number2⟩}`
- `\qtylist[⟨options⟩]{⟨numbers⟩}{⟨unit⟩}`
- `\qtyproduct[⟨options⟩]{⟨numbers⟩}{⟨unit⟩}`
- `\qtyrange[⟨options⟩]{⟨number1⟩}{⟨number2⟩}{⟨unit⟩}`
- `\complexnum[⟨options⟩]{⟨number⟩}`
- `\complexqty[⟨options⟩]{⟨number⟩}{⟨unit⟩}`
- `\sisetup{⟨options⟩}`
- `\tablenum[⟨options⟩]{⟨number⟩}`

加上表格环境中用于小数对齐 (decimal alignments) 和单位 (units) 这二者的 s 列类型 (column type)。这些用户宏和列类型设计用于控制外观 (control of appearance) 和智能处理 (intelligent processing) 来排版数字和单位。

处理数字时要理解指数 (exponents), 或者对乘积 (products) 和复杂数字 (complex numbers) 使用附加命令 (additional commands)。

排版的结果	输入的命令
12 345.678 90	<code>\num{12345,67890}</code>
$0.3 \times 10^{45}$	<code>\num{.3e45}</code>
$1 \pm 2i$	<code>\complexnum{1+-2i}</code>
$1.654 \times 2.34 \times 3.430$	<code>\numproduct{1.654 x 2.34 x 3.430}</code>

单位制 (unit system) 可以将给定的单位解释为直接使用的文本 (text) 或基于宏的单位 (macro-based units)。在后一种情况下，可以使用不同的格式。

输入的命令	排版的结果
<code>\unit{kg.m.s^{-1}}</code>	$\text{kg m s}^{-1}$
<code>\unit{\kilogram\metre\per\second}</code>	$\text{kg m s}^{-1}$
<code>\unit[per-mode=symbol]{\kilogram\metre\per\second}</code>	$\text{kg m/s}$
<code>\unit[per-mode=symbol]{\kilogram\metre\per\ampere\per\second}</code>	$\text{kg m/(A s)}$

可以处理简单的列表 (simple lists) 和数字范围 (ranges of numbers)。

输入的命令	排版的结果
<code>\numlist{10;20;30}</code>	10, 20 and 30
<code>\qtylist{0.13;0.67;0.80}{\milli\metre}</code>	0.13 mm, 0.67 mm and 0.80 mm
<code>\numrange{10}{20}</code>	10 to 20
<code>\qtyrange{0.13}{0.67}{\milli\metre}</code>	0.13 mm to 0.67 mm

有多种选项可用于控制宏包的行为 (behavior)。例如，在标准设置 (standard settings) 下，所有文本都以当前的正立数学字体 (upright math font) 进行排版。这可以调整为使用文本模式 (text mode)，遵循周围格式 (surrounding formatting) 的各个方面，等等。类似地，标准设置是基于讲英语的地区 (English-speaking locale)，但可以根据其他地区的传统进行调整。

## 3 使用 siunitx 宏包

### 3.1 数字

---

`\num` `\num[options]{number}`

---

数字 (numbers) 由 `\num` 宏自动格式化 (automatically formatted)。这需要一个可选参数 (optional argument) *options* 和一个强制性参数 (mandatory argument) *number*。*number* 的内容被自动格式化。格式化程序 (formatter) 删除“软 (soft)”空格 (`\,`) 和“硬 (hard)”空格 (`\,` 和 `\,`)，自动识别指数 (使用 `e`、`E`、`d` 或 `D` 作为标准标记)，并添加适当的大数字间距 (spacing of large numbers)。如果需要，在小数标记 (decimal marker) 之前添加前导零 (leading zero)：. 和 , 都被识别为小数标记。

排版的结果	输入的命令
123	<code>\num{123}</code>
1234	<code>\num{1234}</code>
12 345	<code>\num{12345}</code>
0.123	<code>\num{0.123}</code>
0.1234	<code>\num{0,1234}</code>
0.123 45	<code>\num{.12345}</code>
$3.45 \times 10^{-4}$	<code>\num{3.45d-4}</code>
$-10^{10}$	<code>\num{-e10}</code>

请注意，数字在排版之前被解析 (parsed)，这确实会带来性能开销 (performance overhead)(只有在非常大量的数字输入时才会明显)。解析器 (parser) 理解一系列输入语法 (input syntaxes)，如上所示。

---

`\numlist` `\numlist[options]{numbers}`

---

可以使用 `\numlist` 函数处理数字列表 (lists of numbers)。每个 *number* 在一个括号 (brace) 对中 *numbers* 的列表中给出，因为该列表可以有一个灵活的长度 (flexible length)。

排版的结果	输入的命令
10, 30, 50 and 70	<code>\numlist{10;30;50;70}</code>

---

`\numproduct` `\numproduct[ $\langle options \rangle$ ]{ $\langle numbers \rangle$ }`

---

可以使用 `\numproduct` 函数插入数字的乘积 (products of numbers)。此操作与 `\num` 相同, 但在条目 (entries) 之间插入符号 (symbol) 或短语 (phrase)。后者应以 `x` 标记 (tokens) 分隔。

排版的结果	输入的命令
$10 \times 30$	<code>\numproduct{10 x 30}</code>

---

`\numrange` `\numrange[ $\langle options \rangle$ ]{ $\langle number1 \rangle$ }{ $\langle number2 \rangle$ }`

---

可以使用 `\numrange` 函数处理简单的数字范围 (ranges of numbers)。此操作与 `\num` 相同, 但在两个条目 (entries) 之间插入短语 (phrase) 或其他文本 (other text)。

排版的结果	输入的命令
10 to 30	<code>\numrange{10}{30}</code>

## 3.2 角度

---

`\ang` `\ang[ $\langle options \rangle$ ]{ $\langle angle \rangle$ }`

---

可以使用 `\ang` 命令设置角度 (angles)。这个  $\langle angle \rangle$  可以以小数 (decimal number) 或分号分隔的 (semi-colon separated) 度数 (degrees)、分 (minutes) 和秒 (seconds) 等的列表的形式给出, 在本文档中称为“弧格式 (arc format)”。构成一个角度的数字 (numbers) 与其他数字使用相同的系统处理。

排版的结果	输入的命令
$10^\circ$	<code>\ang{10}</code>
$12.3^\circ$	<code>\ang{12.3}</code>
$4.5^\circ$	<code>\ang{4,5}</code>
$1^\circ 2' 3''$	<code>\ang{1;2;3}</code>
$1''$	<code>\ang{;;1}</code>
$10^\circ$	<code>\ang{+10;;}</code>
$0^\circ 1'$	<code>\ang{-0;1;}</code>

### 3.3 单位

---

`\unit` `\unit[<options>]{<unit>}`

---

可以使用 `\unit` 宏对单位的符号 (symbol for a unit) 进行排版: 这样可以完全控制单位的输出格式 (output format)。与 `\num` 宏一样, `\unit` 具有一个可选参数 (optional argument) 和一个强制参数 (mandatory argument)。单位格式化系统 (unit formatting system) 可以接受两种类型的输入。当 *<unit>* 包含文字项 (literal items), 例如字母 (letters) 或数字 (numbers), 然后 `siunitx` 将 `.` 和 `~` 转换为单位间乘积 (inter-unit product), 并正确定位使用 `_` 和 `^` 指定的下标 (subscripts) 和上标 (superscripts)。格式化方法将同时适用于数学模式 (math mode) 和文本模式 (text mode)。

排版的结果	输入的命令
$\text{kg m/s}^2$	<code>\unit{kg.m/s^2}</code>
$\text{g}_{\text{polymer}} \text{mol}_{\text{cat}} \text{s}^{-1}$	<code>\unit{g_{polymer}~mol_{cat}.s^{-1}}</code>

`\unit` 宏的第二种操作模式 (operation mode) 是“解释 (interpreted)”系统, 这里, 为每个单位、si (国际单位制) 多个词头 (prefix) 和幂 (power) 指定一个宏名称。这些信息的输入方法与单位名称 (unit name) 的英文读法非常相似。

输入的命令	排版的结果
<code>\unit{\kilo\gram\metre\per\square\second}</code>	$\text{kg m s}^{-2}$
<code>\unit{\gram\per\cubic\centi\metre}</code>	$\text{g cm}^{-3}$
<code>\unit{\square\volt\cubic\lumen\per\farad}</code>	$\text{V}^2 \text{lm}^3 \text{F}^{-1}$
<code>\unit{\metre\squared\per\gray\cubic\lux}</code>	$\text{m}^2 \text{Gy}^{-1} \text{lx}^3$
<code>\unit{\henry\second}</code>	$\text{H s}$

就其本身而言, 这不如直接方法 (direct method) 方便, 尽管它确实使用了意义 (meaning) 而不是外观 (appearance) 来输入。但是, 该宏包允许您定义新的单位宏 (new unit macros); 还提供了大量预定义的缩写 (pre-defined abbreviations)。更重要的是, 通过为单位 (units) 定义宏而不是文字输入 (literal input), 新的功能 (new functionality) 变得可用。通过改变宏包使用的设置, 相同的输入可以产生各种不同的输出格式。例如, `\per` 宏可以给出倒数幂 (reciprocal powers)、斜线 (slashes), 宏还可以用于将单位 (units) 构造为分数 (fractions)。

---

`\qty` `\qty[options]{number}{unit}`

---

通常, 数字 (numbers) 和单位 (units) 是一起给出的。形式上, 一个量 (quantity) 的值是数字和单位的乘积 (product), 空格 (space) 被视为一个乘法符号 (multiplication sign)。`\qty` 宏结合了 `\num` 和 `\unit` 的功能, 使其既可行又简单。`<number>` 和 `<unit>` 参数的工作原理与 `\num` 和 `\unit` 宏的工作原理完全相同。

输入的命令	排版的结果
<code>\qty[mode = text]{1.23}{J.mol^{-1}.K^{-1}}</code>	1.23 J mol <sup>-1</sup> K <sup>-1</sup>
<code>\qty{.23e7}{\candela}</code>	0.23 × 10 <sup>7</sup> cd
<code>\qty[per-mode = symbol]{1.99}{\per\kilogram}</code>	1.99/kg
<code>\qty[per-mode = fraction]{1,345}{\coulomb\per\mole}</code>	1.345 $\frac{\text{C}}{\text{mol}}$

可以将单位宏 (unit macros) 设置为在 `\qty` 和 `\unit` 函数之外可用。这不是标准行为 (standard behaviour), 因为存在名称冲突 (name clashes) 的风险 (例如, `\day` 是一个 T<sub>E</sub>X 原语, 多个宏包定义了 `\degree`)。使用“独立 (stand alone)”单位 (units) 的详细信息见第 4.9 节。

---

`\qtylist` `\qtylist[options]{numbers}{unit}`

---

可以使用 `\qtylist` 函数处理带有单位的数字列表。此函数的行为类似于 `\numlist`, 但是在每个数字后面添加了单位。

排版的结果	输入的命令
10 m, 30 m and 45 m	<code>\qtylist{10;30;45}{\metre}</code>

---

`\qtyproduct` `\qtyproduct[options]{numbers}{unit}`

---

可以使用 `\qtyproduct` 函数处理带单位的数 (numbers with units) 的乘积的运行 (runs)。此函数的行为类似于 `\numproduct`, 但在每个数字上添加了一个单位。

排版的结果	输入的命令
10 m × 30 m × 45 m	<code>\qtyproduct{10 x 30 x 45}{\metre}</code>

---

`\qtyrange` `\qtyrange[\langle options \rangle]{\langle number1 \rangle}{\langle number2 \rangle}{\langle unit \rangle}`

---

可以使用 `\qtyrange` 函数处理带单位的数字范围 (ranges of numbers)。此函数的行为类似于 `\numrange`，但在每个数字上添加了一个单位。

排版的结果	输入的命令
10 m to 30 m	<code>\qtyrange{10}{30}{\metre}</code>

使用单个命令输入的列表 (lists)、乘积 (products) 和数量范围 (ranges of quantities) 可以一起调整。这些命令旨在允许对相关值进行一致的格式化：因此，它们对所有值应用单个单位 (single unit)。当使用数值 (numerical values) 调整时，这一点尤其值得注意。

### 3.4 复数和数量

---

`\complexnum` `\complexnum[\langle options \rangle]{\langle number \rangle}`

---

排版复数 (complex number)，它可以是笛卡尔形式 (Cartesian form) 的  $a + bi$  或  $a + ib$ ，也可以是极坐标形式 (polar form) 的  $r:\theta$ 。在其他方面，数字部分 (numerical parts) 的处理与标准 `\num` 命令相同。

---

`\complexqty` `\complexqty[\langle options \rangle]{\langle number \rangle}{\langle unit \rangle}`

---

排版复数 (complex number)，它可以是笛卡尔形式 (Cartesian form) 的  $a + bi$  或  $a + ib$ ，也可以是极坐标形式 (polar form) 的  $r:\theta$ 。在其他方面，数字部分 (numerical parts) 的处理与标准 `\qty` 命令相同。

### 3.5 单位的宏

该宏包始终使用宏名称 (macro names) 定义基本的 si (国际单位制) 的单位。这包括 si 的基本单位 (base units)、具有特殊名称的导出单位 (derived units) 和词头 (prefixes)。少数乘方 (powers) 也被赋予预定义的名称 (pre-defined names)。si 中单位的详细信息可在线获取 [1]。

总是定义七个 si 基本单位 (表 1)。此外，宏 `\meter` 可作为 `\metre` 的别名，为用户的美国拼写。基本单位 (base units) 的详细信息见《si 手册 (si Brochure)》。

表 1: 国际单位制 (si) 的七个基本单位 (base units)

基本量的名称	单位名称 (中)	单位名称 (英)	宏名称	符号
电流	安 [培]	ampere	\ampere	A
发光强度	坎 [德拉]	candela	\candela	cd
热力学温度	开 [尔文]	kelvin	\kelvin	K
质量	千克 (公斤)	kilogram	\kilogram	kg
长度	米	metre	\metre	m
物质的量	摩 [尔]	mole	\mole	mol
时间	秒	second	\second	s

si 还列出了一些具有特殊名称和符号的单位：如表 2 所示。

除了官方的 si 单位外，siunitx 还为一些单位提供宏，这些单位虽然不是 si 单位，但可以在 si 中使用。表 3 列出了“获接纳的 (accepted)”单位。`\percent` 命令也可用于单位：这在手册 (Brochure) 第 5.4.7 节详细说明了 si 中是可以接受的。

除了单位本身，siunitx 还为所有 si 词头 (prefixes) 提供预定义的宏 (表 4)。拼写“\deka”是为美国用户提供的，可以替代 `\deca`。

少量预定义的幂 (powers) 作为宏提供。`\square` 和 `\cubic` 用于单位之前，`\squared` 和 `\cubed` 位于单位之后。

排版的结果	输入的命令
$\text{Bq}^2$	<code>\unit{\square\becquerel}</code>
$\text{J}^2 \text{lm}^{-1}$	<code>\unit{\joule\squared\per\lumen}</code>
$\text{lx}^3 \text{V T}^3$	<code>\unit{\cubic\lux\volt\tesla\cubed}</code>

可以使用 `\tothe` 和 `\raiseto` 宏一次性插入泛型幂 (generic powers)。以下是用于接受参数的单位的唯一宏：

排版的结果	输入的命令
$\text{H}^5$	<code>\unit{\henry\tothe{5}}</code>
$\text{rad}^{4.5}$	<code>\unit{\raiseto{4.5}\radian}</code>

表 2: si 中具有特殊名称和符号的一贯导出单位 (coherent derived units)

量的名称	单位名称 (中)	单位名称 (英)	宏名称	符号
[放射性] 活度	贝可 [勒尔]	becquerel	\becquerel	Bq
力	牛 [顿]	newton	\newton	N
摄氏温度	摄氏度	degree Celsius	\degreeCelsius	°C
电阻	欧 [姆]	ohm	\ohm	$\Omega$
电荷 [量]	库 [仑]	coulomb	\coulomb	C
压力, 压强, 应力	帕 [斯卡]	pascal	\pascal	Pa
电容	法 [拉]	farad	\farad	F
平面角	弧度	radian	\radian	rad
吸收剂量, 比授 [予] 能, 比释动能	戈 [瑞]	gray	\gray	Gy
电导	西 [门子]	siemens	\siemens	S
频率	赫 [兹]	hertz	\hertz	Hz
剂量当量	希 [沃特]	sievert	\sievert	Sv
电感	亨 [利]	henry	\henry	H
立体角	球面度	steradian	\steradian	sr
能 [量], 功, 热量	焦 [尔]	joule	\joule	J
磁通 [量] 密度, 磁感应强度	特 [斯拉]	tesla	\tesla	T
光通量	流 [明]	lumen	\lumen	lm
电压, 电动势, 电位, (电势)	伏 [特]	volt	\volt	V
催化活性, 酶活力	卡 [塔尔]	katal	\katal	kat
功率, 辐 [射能] 通量	瓦 [特]	watt	\watt	W
光 [照度]	勒 [克斯]	lux	\lux	lx
磁 [通量]	韦 [伯]	weber	\weber	Wb

表 3: 获接纳的能与国际单位制 (si) 一起使用的非 si 单位

量的名称	单位名称 (中)	单位名称 (英)	宏名称	符号
长度	天文单位距离	astronomicalunit	<code>\astronomicalunit</code>	au
声压级	贝尔	bel	<code>\bel</code>	B
原子质量	道尔顿	dalton	<code>\dalton</code>	Da
时间	日, 天	day	<code>\day</code>	d
声压级	分贝	decibel	<code>\decibel</code>	dB
平面角, (角度)	度	degree	<code>\degree</code>	°
能	电子伏特	electronvolt	<code>\electronvolt</code>	eV
地积	公顷	hectare	<code>\hectare</code>	ha
时间	[小] 时	hour	<code>\hour</code>	h
体积, 容积	升	litre	<code>\litre</code>	L
			<code>\liter</code>	L
平面角, (角度)	分 (平面角)	minute (plane angle)	<code>\arcminute</code>	'
时间	分	minute (time)	<code>\minute</code>	min
平面角, (角度)	秒 (平面角)	second (plane angle)	<code>\arcsecond</code>	"
场量之比	奈培	neper	<code>\neper</code>	Np
质量	吨	tonne	<code>\tonne</code>	t

使用 `\per` 宏表示倒数幂 (reciprocal powers)。这仅适用于以下单位 (next unit), 除非启用了 `sticky-per` (粘着每) 选项。

排版的结果	输入的命令
$\text{J mol}^{-1} \text{K}^{-1}$	<code>\unit{\joule\per\mole\per\kelvin}</code>
$\text{J mol}^{-1} \text{K}$	<code>\unit{\joule\per\mole\kelvin}</code>
$\text{H}^{-5}$	<code>\unit{\per\henry\tothe{5}}</code>
$\text{Bq}^{-2}$	<code>\unit{\per\square\becquerel}</code>

表 4: 国际单位制词头 (si prefixes)

词头名称 (中)	词头名称 (英)	命令	词头符号	幂
幺 [科托]	yocto	\yocto	y	-24
仄 [普托]	zepto	\zepto	z	-21
阿 [托]	atto	\atto	a	-18
飞 [母托]	femto	\femto	f	-15
皮 [可]	pico	\pico	p	-12
纳 [诺]	nano	\nano	n	-9
微	micro	\micro	μ	-6
毫	milli	\milli	m	-3
厘	centi	\centi	c	-2
分	deci	\deci	d	-1
十	deca	\deca	da	1
百	hecto	\hecto	h	2
千	kilo	\kilo	k	3
兆	mega	\mega	M	6
吉 [咖]	giga	\giga	G	9
太 [拉]	tera	\tera	T	12
拍 [它]	peta	\peta	P	15
艾 [可萨]	exa	\exa	E	18
泽 [它]	zetta	\zetta	Z	21
尧 [它]	yotta	\yotta	Y	24

对于泛型幂 (generic powers), 泛型限定符 (generic qualifiers) 也可以使用 \of 函数:

排版的结果	输入的命令
$\text{kg}_{\text{metal}}$	<code>\unit{\kilogram\of{metal}}</code>
$0.1 \text{ mmol}(\text{cat}) \text{ kg}(\text{prod})^{-1}$	<code>\qty[qualifier-mode = bracket]{0.1}\%</code> <code>{\milli\mole\of{cat}\per\kilogram\of{prod}}</code>

加载 cancel 宏包后, 可以使用 \cancel 宏 “取消 (cancel out)” 单位。这适用于以下单位 (next unit), 方式与词头 (prefix) 类似。 \highlight 宏也可用于选择性地为单位着色

(color units)。 `\cancel` 和 `\highlight` 都超出了单位的正常语义 (semantic meaning) 范围，但在某些情况下可能有用，因此提供了它们。

排版的结果	输入的命令
$\frac{\cancel{\text{kg}}\text{m}}{\cancel{\text{kg}}\text{s}}$	<code>\unit[per-mode = fraction]{\cancel %  \kilogram\metre\per\cancel\kilogram\per\second}</code>
$\textcolor{red}{\text{kg}}\text{ms}^{-1}$	<code>\unit{\highlight{red}\kilogram\metre\per\second}</code>
$\textcolor{blue}{\text{kg}}\textcolor{purple}{\text{ms}}^{-1}$	<code>\unit[unit-color = purple]{ %  \highlight{blue}\kilogram\metre\per\second}</code>

### 3.6 单位的缩写

除了“全名 (full names)”之外，siunitx 还加载了一组 si 单位的缩写版本 (abbreviated versions)(表 5)。标准的 siunitx 设置只在 `\unit` 和 `\qty` 函数的范围内创建这些缩写，这意味着不会发生冲突 (例如与标准的 `\pm` 符号)。

表 5: 单位的缩写 (unit abbreviations)

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
质量	飞克	femtogram	<code>\fg</code>	fg
质量	皮克	picogram	<code>\pg</code>	pg
质量	纳克	nanogram	<code>\ng</code>	ng
质量	微克	microgram	<code>\ug</code>	μg
质量	毫克	milligram	<code>\mg</code>	mg

续下一页

接上一页

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
质量	克	gram	\g	g
质量	千克, 公斤	kilogram	\kg	kg
长度	皮米	picometre	\pm	pm
长度	纳米	nanometre	\nm	nm
长度	微米	micrometre	\um	μm
长度	毫米	millimetre	\mm	mm
长度	厘米	centimetre	\cm	cm
长度	分米	decimetre	\dm	dm
长度	米	metre	\m	m
长度	千米, 公里	kilometre	\km	km
时间	阿秒	attosecond	\as	as
时间	飞秒	femtosecond	\fs	fs
时间	皮秒	picosecond	\ps	ps
时间	纳秒	nanosecond	\ns	ns
时间	微秒	microsecond	\us	μs
时间	毫秒	millisecond	\ms	ms
时间	秒	second	\s	s
物质的量	飞摩尔	femtomole	\fmol	fmol
物质的量	皮摩尔	picomole	\pmol	pmol
物质的量	纳摩尔	nanomole	\nmol	nmol
物质的量	微摩尔	micromole	\umol	μmol
物质的量	毫摩尔	millimole	\mmol	mmol
物质的量	摩尔	mole	\mol	mol

续下一页

接上一页

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
物质的量	千摩尔	kilomole	\kmol	kmol
电流	皮安	picoampere	\pA	pA
电流	纳安	nanoampere	\nA	nA
电流	微安	microampere	\uA	μA
电流	毫安	milliampere	\mA	mA
电流	安	ampere	\A	A
电流	千安	kiloampere	\kA	kA
体积, 容积	微升	microlitre	\u1	μL
体积, 容积	毫升	millilitre	\m1	mL
体积, 容积	升	litre	\1	L
体积, 容积	百升	hectolitre	\h1	hL
体积, 容积	微升	microliter	\uL	μL
体积, 容积	毫升	milliliter	\mL	mL
体积, 容积	升	liter	\L	L
体积, 容积	百升	hectoliter	\hL	hL
频率	毫赫	millihertz	\mHz	mHz
频率	赫	hertz	\Hz	Hz
频率	千赫	kilohertz	\kHz	kHz
频率	兆赫	megahertz	\MHz	MHz
频率	吉赫	gigahertz	\GHz	GHz
频率	太赫	terahertz	\THz	THz
力	毫牛	millinewton	\mN	mN
力	牛	newton	\N	N

续下一页

接上一页

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
力	千牛	kilonewton	\kN	kN
力	兆牛	meganewton	\MN	MN
压力, 压强, 应力	帕	pascal	\Pa	Pa
压力, 压强, 应力	千帕	kilopascal	\kPa	kPa
压力, 压强, 应力	兆帕	megapascal	\MPa	MPa
压力, 压强, 应力	吉帕	gigapascal	\GPa	GPa
电阻	毫欧	milliohm	\mohm	mΩ
电阻	千欧	kilohm	\kohm	kΩ
电阻	兆欧	megohm	\Mohm	MΩ
电压, 电动势, 电位, (电势)	皮伏	picovolt	\pV	pV
电压, 电动势, 电位, (电势)	纳伏	nanovolt	\nV	nV
电压, 电动势, 电位, (电势)	微伏	microvolt	\uV	μV
电压, 电动势, 电位, (电势)	毫伏	millivolt	\mV	mV
电压, 电动势, 电位, (电势)	伏	volt	\V	V
电压, 电动势, 电位, (电势)	千伏	kilovolt	\kV	kV
功率, 辐通量	瓦	watt	\W	W
功率, 辐通量	纳瓦	nanowatt	\nW	nW
功率, 辐通量	微瓦	microwatt	\uW	μW
功率, 辐通量	毫瓦	milliwatt	\mW	mW
功率, 辐通量	千瓦	kilowatt	\kW	kW
功率, 辐通量	兆瓦	megawatt	\MW	MW
功率, 辐通量	吉瓦	gigawatt	\GW	GW
能, 功, 热量	焦	joule	\J	J

续下一页

接上一页

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
能, 功, 热量	微焦	microjoule	\uJ	$\mu\text{J}$
能, 功, 热量	毫焦	millijoule	\mJ	mJ
能, 功, 热量	千焦	kilojoule	\kJ	kJ
能	电子伏特	electronvolt	\eV	eV
能	毫电子伏	millielectronvolt	\meV	meV
能	千电子伏	kiloelectronvolt	\keV	keV
能	兆电子伏	megaelectronvolt	\MeV	MeV
能	吉电子伏	gigaelectronvolt	\GeV	GeV
能	太电子伏	teraelectronvolt	\TeV	TeV
能	千瓦时	kilowatt hour	\kWh	kWh
电容	法	farad	\F	F
电容	飞法	femtofarad	\fF	fF
电容	皮法	picofarad	\pF	pF
电容	纳法	nanofarad	\nF	nF
电容	微法	microfarad	\uF	$\mu\text{F}$
电感	亨	henry	\H	H
电感	飞亨	femtohenry	\fH	fH
电感	皮亨	picohenry	\pH	pH
电感	纳亨	nanohenry	\nH	nH
电感	毫亨	millihenry	\mH	mH
电感	微亨	microhenry	\uH	$\mu\text{H}$
电荷	库	coulomb	\C	C
电荷	纳库	nanocoulomb	\nC	nC

续下一页

接上一页

量的名称	单位名称 (中)	单位名称 (英)	缩写	符号
电荷	毫库	millicoulomb	\mC	mC
电荷	微库	microcoulomb	\uC	$\mu$ C
热力学温度	开	kelvin	\K	K
声压级	分贝	decibel	\dB	dB

`bit` 二进制数据 (binary data) 以位 (bit) 和字节 (byte) 为单位表示。这些词头 (prefixes) `byte` 通常使用 2 的幂 (powers), 而不是 si 词头使用的 10 的幂。由于这些二进制词头与 si 词头密切相关, 因此它们由 `siunitx` 定义。

表 6: 二进制词头 (Binary prefixes)

词头	命令	符号	幂
kibi	\kibi	Ki	10
mebi	\mebi	Mi	20
gibi	\gibi	Gi	30
tebi	\tebi	Ti	40
pebi	\pebi	Pi	50
exbi	\exbi	Ei	60
zebi	\zebi	Zi	70
yobi	\yobi	Yi	80

### 3.7 创建新的宏

必须先定义单位 (unit) 的各种宏组件 (macro components), 然后才能使用它们。该宏包提供了许多常见的定义 (common definitions), 但也可以使用新的定义 (new definitions)。由于逻辑单位 (logical unit) 的定义在单个文档中应该保持不变, 所以这些创建函数 (creation functions) 都仅限于前言 (preamble)。

---

`\DeclareSIUnit` `\DeclareSIUnit[ $\langle options \rangle$ ]{ $\langle unit \rangle$ }{ $\langle symbol \rangle$ }`

---

使用 `\DeclareSIUnit` 宏生成新的单位 (new units)。这个  $\langle symbol \rangle$  可以包含文字输入 (literal input)、其他单位、多个词头 (prefixes)、幂 (powers) 和 `\per`，但文字文本 (literal text) 不应与单位宏 (unit macros) 混合。单位 (units) 可以使用  $\langle options \rangle$  从 `siunitx` 所理解的通常列表 (usual list) 中创建，并且只应用特定的单位宏。`\qty` 和 `\unit` 的 (第一个) 可选参数可用于覆盖单位的设置 (settings)：例如，`\degree` 单位。

排版的结果	输入的命令
3.1415°	<code>\qty{3.1415}{\degree}</code>

这在宏包中 (实际上) 声明为：

`\DeclareSIUnit[quantity-product = {}]\degree{\text{\textdegree}}`

间距 (spacing) 在使用时仍然可以改变：

输入的命令	排版的结果
<code>\qty{67890}{\degree}</code>	67 890°
<code>\qty[quantity-product = \,]{67890}{\degree}</code>	67 890°

在加载 `siunitx` 后，可以使用 `\DeclareSIUnit` 更改预定义单位 (pre-defined unit) 的含义。这 will 用新版本覆盖原始定义 (original definition)。

---

`\DeclareSIPrefix` `\DeclareSIPrefix{ $\langle prefix \rangle$ }{ $\langle symbol \rangle$ }{ $\langle powers-ten \rangle$ }`

---

10 的标准 si 幂 (standard si powers) 由宏包定义，如上所述。但是，用户可以使用 `\DeclareSIPrefix` 定义新词头 (new prefixes)。例如，`\kilo` 的定义为：

`\DeclareSIPrefix\kilo{k}{3}`

---

`\DeclareSIPower` `\DeclareSIPower{ $\langle symbol-before \rangle$ }{ $\langle symbol-after \rangle$ }{ $\langle power \rangle$ }`

---

这个函数创建了两个符号 (symbols)，一个在单位前 (before a unit) 使用，另一个在单位后 (after a unit) 使用，它们都等价于  $\langle power \rangle$ 。例如，您可以将

`\DeclareSIPower\quartic\tothefourth{4}`

与随后在文档中使用的函数 (functions) 一起使用，如下：

排版的结果	输入的命令
$\text{kg}^4$	<code>\unit{\kilogram\tothefourth}</code>
$\text{m}^4$	<code>\unit{\quartic\metre}</code>

`\DeclareSIQualifier` `\DeclareSIQualifier{<qualifier>}{<symbol>}`

遵循其他宏的语法 (syntax)，可以使用 `\DeclareSIQualifier` 命令创建限定符 (qualifiers)。与单位的其他部分不同，没有预定义的限定符 (pre-defined qualifiers)。因此，创建这些文件完全取决于用户。例如，为了确定使用特定催化剂 (particular catalyst) 时产生的产品质量 (mass)，前言可以包含：

`\DeclareSIQualifier\polymer{pol}`  
`\DeclareSIQualifier\catalyst{cat}`

然后在正文 (body) 中文档可以读取

输入的命令	排版的结果
<code>\qty{1.234}{\gram\polymer\per\mole\catalyst\per\hour}</code>	$1.234 \text{ g}_{\text{pol}} \text{ mol}_{\text{cat}}^{-1} \text{ h}^{-1}$

### 3.8 表格的材料

表格内容 (tabular content) 中的数字对齐 (aligning numbers) 由新的列类型 (column type) `S` 列处理。这种新的列类型可以使用一系列不同的策略 (strategies) 对齐材料 (material)，目的是灵活的输出而不需要改变输入。标准的方法是将小数标记 (decimal marker) 放置在单元格中心的数字中，并适当地对齐材料 (表 7)。

`S` 列将尝试自动检测应该放置在数字之前或之后的材料，并保持数字数据的对齐 (表 8)。如果材料可能被误认为一个数字的一部分，它应该被大括号 (braces) 保护。它还将检测单元格中 `\color` 的使用，并覆盖 `siunitx` 应用的任何常规颜色 (general color)。

输入的代码：

表 7: 排版的结果 (s 列类型的标准行为)

<code>\begin{table}</code>	
<code>\begin{tabular}{@{}S@{}}</code>	一些值
<code>\hline{1.2pt}</code>	2.3456
<code>{\Heiti 一些值} \\</code>	34.2345
<code>\hline{0.7pt}</code>	-6.7835
<code>2.3456 \\</code>	90.473
<code>34.2345 \\</code>	5642.5
<code>-6.7835 \\</code>	$1.2 \times 10^3$
<code>90.473 \\</code>	$10^4$
<code>5642.5 \\</code>	
<code>1.2e3 \\</code>	
<code>e4 \\</code>	
<code>\hline{1.2pt}</code>	
<code>\end{tabular}</code>	
<code>\end{table}</code>	

输入的代码：

表 8: 排版的结果 (s 列中周围材料的检测)

<code>\begin{table}</code>	一些值
<code>\begin{tabular}{%}</code>	12.34
<code>@{}S[color = orange]{}</code>	975.31
<code>\hline{1.2pt}</code>	44.268 <sup>a</sup>
<code>{\Heiti 一些值} \\</code>	
<code>\hline{0.7pt}</code>	
<code>12.34 \\</code>	
<code>\color{purple} 975,31 \\</code>	
<code>44.268 \textsuperscript{\emph{a}} \\</code>	
<code>\hline{1.2pt}</code>	
<code>\end{tabular}</code>	
<code>\end{table}</code>	

---

`\tablenum` `\tablenum[options]{number}`

---

在更复杂的表格中，可能需要在 `\multicolumn` 或 `\multirow`<sup>2</sup> 的参数中对齐数字。`\tablenum` 函数可用于在这些情况下实现对齐：这实际上是 S 列的宏版本 (macro version)(表 9)。

```
\begin{table}[H]
  \caption{\ {\Heiti 使用} \cs{tablenum}\ {\Heiti 宏控制复杂的对齐}}%
  \label{tab:tablenum}}
  \begin{tabular}{@{}lr@{}}
    \hline{1.2pt}
    {\Heiti 表头} & {\Heiti 表头} \\
    \hline{0.7pt}
    信息 & 更多信息 \\
    信息 & 更多信息 \\
    \multicolumn{2}{c}{\tablenum[table-format = 4.4]{12,34}} \\
    \multicolumn{2}{c}{\tablenum[table-format = 4.4]{333.5567}} \\
    \multicolumn{2}{c}{\tablenum[table-format = 4.4]{4563.21}} \\
    \hline{1.2pt}
  \end{tabular}
  \hspace{5em}%
  \begin{tabular}{@{}lr@{}}
    \hline{1.2pt}
    {\Heiti 表头} & {\Heiti 表头} \\
    \hline{0.7pt}
    \multirow{2}{*{\tablenum{88,999}}} & aaa \\
    & bbb \\
    \multirow{2}{*{\tablenum{33,435}}} & ccc \\
    & ddd \\
    \hline{1.2pt}
  \end{tabular}
\end{table}
```

---

<sup>2</sup>由 `multirow` 宏包提供

表 9: 使用 \tablenum 宏控制复杂的对齐

表头	表头	表头	表头
信息	更多信息	88.999	aaa
信息	更多信息		bbb
	12.34		ccc
	333.5567	33.435	ddd
	4563.21		

## 4 宏包的控制选项

### 4.1 键-值控制系统

该宏包使用了一系列不同的键类型 (key types):

**Choice** 采用有限数量的选项 (choices), 分别针对每个键 (key) 进行描述。

**Integer** 需要一个数字作为参数。

**Length** 需要一个长度 (length), 可以是文本值 (literal value) 例如 2.0cm, 也可以是 L<sup>A</sup>T<sub>E</sub>X 长度。

**Literal** 一种键, 它使用直接给出的值来检查输入或输出。

**Macro** 需要一个宏, 该宏可能需要一个参数。

**Math** 类似于 **literal** 选项, 但无论其他 **siunitx** 设置如何, 输入始终在数学模式 (math mode) 下使用。因此, 对于文本模式 (text-mode), 只能将输入置于 `\text` 宏的参数内。

**Meta** 这些选项实际上应用了许多其他选项。

**Switch** 这些是开关 (on-off switches), 可以识别 **true** 和 **false**。只给出键名 (key name) 也可以打开键。

选项名称表 (tables of option names) 使用这些描述来指示应如何使用键。

### 4.2 打印

**siunitx** 宏软件包可以独立于周围材料 (surrounding material) 控制用于打印输出 (print output) 的字体。字体的哪些方面遵循环境的哪些方面受到表 10 中详述的设置范围 (range of setting) 的影响。

**mode (env.)** **mode** 选项决定打印输出时 **siunitx** 是使用数学模式 (math mode) 还是文本模式 (text mode)。选项包括 **match**、**math**、**text**。**match** 设置意味着打印使用当前模式不变, 而 **math** 和 **text** 选择适当的 T<sub>E</sub>X 模式。在数学模式和文本模式中可以使用不同的字体, 这将突出显示 (highlight) 差异 (difference)。应用的字体设置 (font settings) 也因模式而异。**number-mode (env.)** 除了整体设置 (overall setting) 外, 还可以使用 **number-mode** 选项将模式应用于数字 (numbers), 使用 **unit-mode (env.)** 将模式应用于单位 (units)。

表 10: 打印选项 (Print options)

选项名称	含义	类型	默认
color	颜色	Literal	$\langle none \rangle$
mode	模式	Choice	math
number-color	数字颜色	Literal	$\langle none \rangle$
number-mode	数字模式	Choice	math
propagate-math-font	变形数学字体	Switch	false
reset-math-version	重置数学版本	Switch	true
reset-text-family	重置文本族	Switch	true
reset-text-series	重置文本序列	Switch	true
reset-text-shape	重置文本形状	Switch	true
text-family-to-math	文本族到数学	Switch	false
text-font-command	文本字体命令	Literal	$\langle none \rangle$
text-subscript-command	文本下标命令	Literal	$\backslash\text{textsubscript}$
text-superscript-command	文本上标命令	Literal	$\backslash\text{textsuperscript}$
text-series-to-math	文本序列到数学	Switch	false
unit-color	单位颜色	Literal	$\langle none \rangle$
unit-mode	单位模式	Choice	math

`reset-text-family (env.)` 在文本模式 (text mode) 中打印时, 将应用 `reset-text-family` (重置文本族)、`reset-text-series (env.)` `reset-text-series` (重置文本序列) 和 `reset-text-shape (env.)` `reset-text-shape` (重置文本形状) 选项。当这些选项处于活动状态时, siunitx 会在打印时重置相关的字体选择轴属性 (font selection axis property): 标准字体设置 (standard font setting) 是直立中重 (upright mid-weight) 罗马字体 (`\upshape \mdseries \rmfamily`)。

结果	命令
1234	<code>\sisetup{mode=text}{\itshape \num{1234}}</code>
1234	<code>\sisetup{mode=text}{\bfseries \num{1234}}</code>
1234	<code>\sisetup{mode=text}{\sffamily \num{1234}}</code>
1234	<code>\sisetup{mode=text,reset-text-family=false,reset-text-series=false, reset-text-shape=false}{\itshape \num{1234}}</code>
1234	<code>\sisetup{mode=text,reset-text-family=false,reset-text-series=false, reset-text-shape=false}{\bfseries \num{1234}}</code>
1234	<code>\sisetup{mode=text,reset-text-family=false,reset-text-series=false, reset-text-shape=false}{\sffamily \num{1234}}</code>

在数学模式 (math mode) 中, L<sup>A</sup>T<sub>E</sub>X 使用的字体是“不变 (invariant)”的, 这反映在可用选项中。使用标准设置 (standard settings), 在数学模式中, 打印使用标准数学字体和版本 `propagate-math-font (env.)` (权重)。 `propagate-math-font` (变形数学字体) 选项可用于将常用数学字体 (prevailing `reset-math-version (env.)` math font) 应用于打印材料 (printed material)。 `reset-math-version` (重置数学版本) 设置控制是否重置数学版本 (math version)。注意, 数学版本通常用于设置“粗体数学 (bold math)”, 但也可以用于其他效果, 例如所有无衬线数学 (sanserif math)。

排版的结果	输入的命令
kg	<code>{\boldmath \unit{\kilogram}}</code>
kg	<code>{\sansmath \$\unit{\kilogram}\$}</code>
kg	<code>{\mathsf{\unit{\kilogram}}\$}</code>
kg	<code>\sisetup{propagate-math-font=true,reset-math-version=false} {\boldmath \unit{\kilogram}}</code>
kg	<code>\sisetup{propagate-math-font=true,reset-math-version=false} {\sansmath \$\unit{\kilogram}\$}</code>
kg	<code>\sisetup{propagate-math-font=true,reset-math-version=false} {\mathsf{\unit{\kilogram}}\$}</code>

`text-family-to-math (env.)` 可以使用选项 `text-family-to-math` (文本族到数学) 和选项 (文本序列到数学) 即 `text-series-to-math (env.)` `text-series-to-math` 来匹配 (尽可能) 数学模式输出到周围的文本 (surrounding text)。这些选项通过检测当前文本设置 (current text settings) 并在数学模式中进行适当选择来工作。

排版的结果	输入的命令
kg	<code>\sffamily \unit{\kilogram}</code>
kg	<code>{\bfseries \$\unit{\kilogram}\$}</code>
kg	<code>\sisetup{text-family-to-math=true,text-series-to-math=true}</code> <code>{\sffamily \unit{\kilogram}}</code>
<b>kg</b>	<code>\sisetup{text-family-to-math=true,text-series-to-math=true}</code> <code>{\bfseries \$\unit{\kilogram}\$}</code>

在某些情况下，当以文本模式 (text mode) 打印时，可能需要使用非标准字体命令 (non-standard font command)。例如，这可以用于在仍然使用文本模式的情况下从旧样式 (old-style) 切换到行号 (lining numbers)。这可以通过设置 `text-font-command` (文本字体命令) 来实现。例如，此文档使用文本模式中的旧样式数字 (old-style numbers) 作为标准，可以通过选择不具有这些数字特性 (feature) 的字体变体 (font variant) 来重写这些数字。

排版的结果	输入的命令
123 456 789 kV cm <sup>-1</sup>	<code>\sisetup{number-mode=text}\qty{123456789}</code> <code>{\kilo\volt\per\centi\metre}</code>
123 456 789 kV cm <sup>-1</sup>	<code>\sisetup{number-mode=text,text-font-command=</code> <code>\fontfamily{pplx}\selectfont}\qty{123456789}</code> <code>{\kilo\volt\per\centi\metre}</code>

`text-subscript-command` (env.) 在大多数情况下，命令 `\textsubscript` (文本下标) 适用于在文本模式中创建下标材料 (subscript material)，命令 `\textsuperscript` (文本上标) 适用于在文本模式中创建上标材料 (superscript material)。然而，当加载 `realscripts` 宏包时，输出可能不符合期望。因此，这两个命令可用于调整结果。它们也可以用于非标准效果 (non-standard effects)，例如在这里为下标添加颜色。

排版的结果	输入的命令
kg <sub>polymer</sub>	<code>\sisetup{unit-mode=text}\unit{\kg\of{polymer}}</code>
kg <sub>polymer</sub>	<code>\newcommand*\mysubscript[1]{\textsubscript{\textcolor{blue}</code> <code>{#1}}}\sisetup{unit-mode=text,text-subscript-command=</code> <code>\mysubscript}\unit{\kg\of{polymer}}</code>

`color (env.)` 可以使用 `color` (颜色) 选项设置打印输出的颜色 (color of printed output)。如果没有指定颜色，打印的颜色将跟随周围的文本。相反，当给定特定颜色时，无论周围环境如何，都会使用该颜色。与 `mode` (模式) 一样，`color` (颜色) 设置也可以独立应用于数字 `unit-color (env.)` (numbers) 和单位 (units)。

排版的结果	输入的命令
Some text	<code>\color{red} Some text \\\</code>
4 kg	<code>\qty{4}{\kilogram} \\\</code>
More text	<code>More text \\\</code>
4 kg	<code>\qty[color=blue]{4}{\kilogram} \\\</code>
Still red here!	<code>Still red here!</code>

### 4.3 解析数字

该宏包使用复杂的解析系统 (parsing system) 来理解数字。这允许 `siunitx` 执行一系列格式化，如稍后所述。所有的输入选项都采用文字标记列表 (lists of literal tokens)，该文字标记列表汇总在表 11 中。

表 11: 用于数字解析 (number parsing) 的选项

选项的名称	含义	类型	默认
<code>evaluate-expression</code>	求值表达式	Switch	<code>false</code>
<code>expression</code>	表达式	Literal	<code>#1</code>
<code>input-close-uncertainty</code>	输入关不确定性	Literal	<code>)</code>
<code>input-comparators</code>	输入比较符	Literal	<code>&lt;=&gt;\approx\ge\geq</code> <code>\gg\le\leq\ll \sim</code>
<code>input-decimal-markers</code>	输入小数点标记	Literal	<code>.,</code>
<code>input-digits</code>	输入数字	Literal	<code>0123456789</code>
<code>input-exponent-markers</code>	输入指数标记	Literal	<code>dDeE</code>
<code>input-ignore</code>	输入忽略	Literal	<code>\langle none \rangle</code>
<code>input-open-uncertainty</code>	输入开不确定性	Literal	<code>(</code>
<code>input-signs</code>	输入符号	Literal	<code>+-\pm\mp</code>
<code>input-uncertainty-signs</code>	输入不确定性符号	Literal	<code>\pm</code>
<code>parse-numbers</code>	解析数字	Switch	<code>true</code>
<code>retain-explicit-decimal-marker</code>	保留显式小数标记	Switch	<code>false</code>
<code>retain-explicit-plus</code>	保留显式加号	Switch	<code>false</code>
<code>retain-negative-zero</code>	保留负零	Switch	<code>false</code>
<code>retain-zero-uncertainty</code>	保留零不确定性	Switch	<code>false</code>

数量 (number) 的基本部分是数字 (digits)、任意符号 (any sign) 以及整数和小数部分之间的分隔符 (separator)。数字 (digits) 存储在输入选项 `input-digits` (输入数字) 中, `input-decimal-markers` (env.) 任意符号 (any sign) 存储在输入选项 `input-decimal-markers` (输入小数点标记) 中, `input-signs` (env.) 分隔符 (separator) 存储在输入选项 `input-signs` (输入符号) 中。可以使用多个输入小数点标记 (input decimal marker): 它将由宏包转换为适当的输出标记。含指数部分的数字也需要一个指数标记: 这同样来自 `input-exponent-markers` (输入指数标记) 选项中的标记 (env.) 记范围 (range of tokens)。

`input-ignore` (env.) `input-ignore` (输入忽略) 列表中给出的令牌完全由 `siunitx` 传递: 它们将从输入中删除, 无需进一步处理。

`input-comparators` (env.) 除了符号 (signs), `siunitx` 还可以识别比较符 (comparators), 例如 `<`。宏包自动将 `<<`, `>>`, `<=` 和 `>=` 分别转换为 `\ll`, `\gg`, `\le` 和 `\ge`。

排版的结果	输入的命令
$<10$	<code>\num{&lt; 10}</code>
$\gg 5\text{ m}$	<code>\qty{&gt;&gt; 5}{\metre}</code>
$\leq 0.12$	<code>\num{\le 0.12}</code>

在某些领域，通常在数字的主要部分 (main part of the number) 后的括号中给出数字的不确定性 (uncertainty in a number)，例如 “1.234(5)”。用于这种类型输入的打开和关闭符号 (opening and closing symbols) 被设置为 `input-open-uncertainty` (输入开不确定性) 和 `input-close-uncertainty` (输入关不确定性)。或者，不确定性 (uncertainty) 可以作为符号 (sign) 后面的单独部分 (separate part) 给出。哪些符号对该操作有效，取决于 `input-uncertainty-signs` (输入不确定性符号) 选项。与其他符号一样，组合 `+-` 将在内部自动转换为 `\pm`。

排版的结果	输入的命令
$9.99(9)$	<code>\num{9.99(9)}</code>
$9.99(9)$	<code>\num{9.99 +- 0.09}</code>
$9.99(9)$	<code>\num{9.99 \pm 0.09}</code>
$123.0(45)$	<code>\num{123 +- 4.5}</code>
$12.3(60)$	<code>\num{12.3 +- 6}</code>

跨越小数点标记 (decimal marker) 的不确定性 (uncertainties) 可以用或不用 “紧凑 (compact)” 形式的小数点标记给出。这些被代码视为等同的。<sup>3</sup>

排版的结果	输入的命令
$123.4(12)$	<code>\num{123.4(12)}</code>
$123.4(12)$	<code>\num{123.4(1.2)}</code>

也可以给出多种不确定性：这些不确定性要么是短形式的 (short form)，要么是长形式的 (long form)。

<sup>3</sup>宏包的作者喜欢没有小数点标记 (decimal marker) 的形式 (form)，而且形式指南 (formal guidance) 对于哪种表式是正确的模棱两可的。带有小数点标记的表开可以在一些 nist 出版物中看到。

排版的结果	输入的命令
$123.4 \pm 1.2 \pm 4.5$	<code>\num{123.4(12)(45)}</code>
$123.4 \pm 1.2 \pm 4.5$	<code>\num{123.4 \pm 1.2 \pm 4.5}</code>

`parse-numbers (env.)` `parse-numbers` (解析数字) 选项可打开和关闭整个解析系统 (parsing system)。该选项可用的原因有两个。首先, 如果一个文档中的所有数字都“按给定”重新生成 (reproduced), 那么关闭解析器将大大节省所需的处理时间。第二, 它允许在数字中使用任意  $\text{T}_{\text{E}}\text{X}$  代码。如果关闭解析器, 输入将以数学模式打印 (需要 `\text` 保护数字中的任何文本)。

排版的结果	输入的命令
$\sqrt{2}$	<code>\num[parse-numbers = false]{\sqrt{2}}</code>
$\sqrt{3}\text{m}$	<code>\qty[parse-numbers = false]{\sqrt{3}}{\metre}</code>

在标准设置 (standard settings) 中, 数字输入 (numerical input) 按“原样”进行解析, 而无需尝试对其进行数学解释 (interpret it mathematically)。通过启用 `evaluate-expression (env.)` `evaluate-expression` (求值表达式) 选项, 可以用标准的  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}3$  `fpu` 处理输入 (更多信息请参见 `xfp` 宏包)。表达式 (expression) 本身的性质 (nature) 可以使用 `expression (env.)` (表达式) 设置进行调整: 作为标准, 整个输入只需解析 (parsed) 而不做任何更改, 但此设置可用于添加其他步骤 (additional steps)。这种表达式中的输入 (input) 由 `#1` 表示。请注意, `fpu` 对数字使用自己的语法, 最明显的是小数点标记 (decimal marker) 必须是 `.`。

排版的结果	输入的命令
14J	<code>\sisetup{evaluate-expression}\qty{2 + 4 * 3}{\joule}</code>
140J	<code>\sisetup{evaluate-expression} % \qty[expression = 10 * (#1)]{2 + 4 * 3}{\joule}</code>

`retain-explicit-decimal-marker (env.)` 在某些区域, 使用没有小数部分的尾随小数标记 (trailing decimal marker) 来表示整数部分 (integer part) 中的零是有效的。这可以通过 `retain-explicit-decimal-marker` (保留显式小数标记) 选项来启用。对于正数 (positive numbers), 通常不需要包含前导加号 (leading plus sign), 因此在解析时不会保留它们作为标准。`retain-explicit-plus (env.)` (保留显式加号) 选项可用于控制此行为。类似地, 零的不确定性 (uncertainty of zero) `retain-zero-uncertainty (env.)` 通常没有意义, 因此解析器 (parser) 会忽略它。这可以使用 `retain-zero-uncertainty` (保留零不确定性) 选项进行控制。最后, 对于完全为零的值 (entirely zero value), 负号

`retain-negative-zero (env.)` (negative sign) 可能有意义，也可能没有意义：这由 `retain-negative-zero` (保留负零) 选项控制。

排版的结果	输入的命令
10	<code>\num{10.}</code>
10.	<code>\num[retain-explicit-decimal-marker]{10.}</code>
345	<code>\num{+345}</code>
+345	<code>\num[retain-explicit-plus]{+345}</code>
0	<code>\num{-0}</code>
-0	<code>\num[retain-negative-zero]{-0}</code>
12.3	<code>\num{12.3(0)}</code>
12.3(0)	<code>\num[retain-zero-uncertainty]{12.3(0)}</code>

4.4 数字后处理

在排版数字之前，可以执行各种后处理步骤 (post-processing steps)。这包括以系统的方式 (in a systematic way) 从数字中添加或删除信息。表 12 总结了这些选项。

表 12: 数字后处理 (post-processing) 选项

选项的名称	含义	类型	默认值
<code>drop-exponent</code>	丢弃指数	Switch	false
<code>drop-uncertainty</code>	丢弃不确定性	Switch	false
<code>drop-zero-decimal</code>	弃零小数	Switch	false
<code>exponent-mode</code>	指数模式	Switch	input
<code>fixed-exponent</code>	固定指数	Integer	0
<code>minimum-integer-digits</code>	最小整数位数	Integer	0
<code>minimum-decimal-digits</code>	最小小数位数	Integer	0
<code>round-half</code>	舍入半	Choice	up
<code>round-minimum</code>	舍入最小值	Literal	0
<code>round-mode</code>	舍入模式	Choice	none
<code>round-pad</code>	舍入填充	Switch	true
<code>round-precision</code>	舍入精度	Integer	2
<code>round-zero-positive</code>	舍入零正	Switch	true

该宏包可以将数字 (numbers) 转换为科学记数法 (scientific notation)。这是由 `exponent-mode (env.)` `exponent-mode` (指数模式) 选项控制的, 该选项的取值为可以是 `input` (输入)、`fixed` (固定)、`engineering` (工程) 和 `scientific` (科学)。该选项如设置为 `fixed`, 将通过 `fixed-exponent (env.)` `fixed-exponent` (固定指数) 选项来使用指数值 (exponent value); 如设置为 `engineering` 时, 则指数总是 3 的幂 (power of three)。

排版的结果	输入的命令
0.001	<code>\num{0.001}</code>
0.0100	<code>\num{0.0100}</code>
1200	<code>\num{1200}</code>
$1 \times 10^{-3}$	<code>\sisetup{exponent-mode=scientific}\num{0.001}</code>
$1.00 \times 10^{-2}$	<code>\sisetup{exponent-mode=scientific}\num{0.0100}</code>
$1.200 \times 10^3$	<code>\sisetup{exponent-mode=scientific}\num{1200}</code>
$1 \times 10^{-3}$	<code>\sisetup{exponent-mode=engineering}\num{0.001}</code>
$10.0 \times 10^{-3}$	<code>\sisetup{exponent-mode=engineering}\num{0.0100}</code>
$1.200 \times 10^3$	<code>\sisetup{exponent-mode=engineering}\num{1200}</code>
$0.000\,01 \times 10^2$	<code>\sisetup{exponent-mode=fixed,fixed-exponent=2}\num{0.001}</code>
$0.000\,100 \times 10^2$	<code>\sisetup{exponent-mode=fixed,fixed-exponent=2}\num{0.0100}</code>
$12.00 \times 10^2$	<code>\sisetup{exponent-mode=fixed,fixed-exponent=2}\num{1200}</code>

当与设置为零的 `fixed-exponent` (固定指数) 一起使用时, 这可以用来从输入中删除科学记数法 (scientific notation)。

排版的结果	输入的命令
$1.23 \times 10^4$	<code>\num{1.23e4}</code>
12 300	<code>\num[exponent-mode = fixed, fixed-exponent = 0]{1.23e4}</code>

指数模式 (exponent mode) 适用于舍入 (rounding) 之后, 以便舍入的小数位数 (number of decimal) 是那些出现在输出中的小数位数。

`drop-uncertainty (env.)` 使用 `drop-uncertainty` (丢弃不确定性) 选项可以完全抑制不确定性 (uncertainty) `drop-exponent (env.)` 的使用: 这在尝试舍入之前适用。类似地, 可以使用 `drop-exponent` (丢弃指数) 选项来丢弃 (drop) 指数。`drop-exponent` 可以用于抑制指数部分 (转换为固定指数 [fixed exponent] 之后)。

排版的结果	输入的命令
0.01(2)	<code>\num{0.01(2)}</code>
0.01	<code>\num[drop-uncertainty]{0.01(2)}</code>
$0.01 \times 10^3$	<code>\num{0.01e3}</code>
0.01	<code>\num[drop-exponent]{0.01e3}</code>

该宏包可以将数字输入 (numerical input) 舍入为固定数量的有效数字 (significant figures) 或小数位数 (decimal places)。这是由 `round-mode` (舍入模式) 选项控制的, 该选项的取值可以是 `none` (无)、`figures` (数字)、`places` (位数) 和 `uncertainty` (不确定)。启用 `round-precision` (舍入精度) 选项设置使用的数的位数 (number of digits) 即小数位数 (decimal places) 或尾数 (mantissa) 中的有效位数 (significant figures)。可以将 `round-mode` 设置为 `uncertainty` 来对具有不确定性的数字进行舍入。在这种情况下, 首先使用精度 (precision) 来舍入不确定性本身 (到若干数字), 然后再舍入主值 (main value)。

排版的结果	输入的命令
1.23456	<code>\num{1.23456}</code>
14.23	<code>\num{14.23}</code>
0.12345(9)	<code>\num{0.12345(9)}</code>
1.235	<code>\sisetup{round-mode=places,round-precision=3}\num{1.23456}</code>
14.230	<code>\sisetup{round-mode=places,round-precision=3}\num{14.23}</code>
0.12345(9)	<code>\sisetup{round-mode=places,round-precision=3}\num{0.12345(9)}</code>
1.23	<code>\sisetup{round-mode=figures,round-precision=3}\num{1.23456}</code>
14.2	<code>\sisetup{round-mode=figures,round-precision=3}\num{14.23}</code>
0.12345(9)	<code>\sisetup{round-mode=figures,round-precision=3}% \num{0.12345(9)}</code>
0.12345(9)	<code>\sisetup{round-mode=uncertainty,round-precision=1}% \num{0.12345(9)}</code>
0.1235(2)	<code>\sisetup{round-mode=uncertainty,round-precision=1}% \num{0.12345(23)}</code>
0.123(2)	<code>\sisetup{round-mode=uncertainty,round-precision=1}% \num{0.12345(234)}</code>

舍入 (rounding) 可能会将一个短数字 (short number) “扩展 (extend)” 到更多的数字

`round-pad (env.)` (digits) 或符号 (figures): 这由开关 (switch) `round-pad` (舍入填充) 控制, 作为标准 (as standard), `round-pad` 是 `true`。

排版的结果	输入的命令
12.30	<code>\sisetup{round-mode=figures,round-precision=4}\num{12.3}</code>
12.3	<code>\sisetup{round-mode=figures,round-precision=4}% \num[round-pad=false]{12.3}</code>

如果数字的舍入部分 (rounded part of a number) 正好是一半 (half), 有两种常见 `round-half (env.)` 的“打破平局 (breaking the tie)”的方法。方法的选择由 `round-half` (舍入半) 决定, `round-half` 识别选择 `up` 和 `even` (舍入到偶数)。

排版的结果	输入的命令
0.06	<code>\sisetup{round-mode=figures,round-precision=1,% round-half=up}\num{0.055}</code>
0.05	<code>\sisetup{round-mode=figures,round-precision=1,% round-half=up}\num{0.045}</code>
0.06	<code>\sisetup{round-mode=figures,round-precision=1,% round-half=even}\num{0.055}</code>
0.04	<code>\sisetup{round-mode=figures,round-precision=1,% round-half=even}\num{0.045}</code>

有些情况下, 舍入将导致数字达到零。可能希望将这样的结果显示为低于阈值 `round-minimum (env.)` (threshold value)。这可以通过将 `round-minimum` (舍入最小值) 设置为阈值 (threshold value) 来实现。舍入到多个有效数字 (significant figures) 时不会产生任何影响, 因为在这些情况下不可能得到零值。

排版的结果	输入的命令
0.01	<code>\sisetup{round-mode=places}\num{0.0055}</code>
0.00	<code>\sisetup{round-mode=places}\num{0.0045}</code>
0.01	<code>\sisetup{round-mode=places,round-minimum=0.01}\num{0.0055}</code>
<0.01	<code>\sisetup{round-mode=places,round-minimum=0.01}\num{0.0045}</code>

当将负数 (negative numbers) 舍入到固定的位数 (fixed number of places) 时, 可能会产生零值。通常这表示为一个无符号值 (unsigned value), 但在某些情况下保留负号

`round-zero-positive (env.)` (negative sign) 可能是可取的。这种行为可以通过使用 `round-zero-positive` (舍入零正) 开关来控制。

排版的结果	输入的命令
0.00	<code>\sisetup{round-mode=places}\num{-0.001}</code>
-0.00	<code>\sisetup{round-mode=places,round-zero-positive=false}% \num{-0.001}</code>

当小数部分 (decimal part) 为零时，可能需要将小数转换为整数 (integers)。这是使 `drop-zero-decimal (env.)` 用 `drop-zero-decimal` (弃零小数) 选项来设置的，该选项在舍入之后但在设置最小位数 (setting minimum numbers of digits) 之前适用。

排版的结果	输入的命令
2.0	<code>\num{2.0}</code>
2.1	<code>\num{2.1}</code>
2	<code>\sisetup{drop-zero-decimal}\num{2.0}</code>
2.1	<code>\sisetup{drop-zero-decimal}\num{2.1}</code>

`minimum-decimal-digits`      `minimum-decimal-digits` (最小小数位数) 和 `minimum-integer-digits` (最小整数位数) 选项可用于将数字填充 (pad) 到给定大小 (given size)。这与任何舍入无关。

`minimum-integer-digits (env.)`

`(env.)`

排版的结果	输入的命令
123	<code>\num{123}</code>
123	<code>\num[minimum-integer-digits=2]{123}</code>
0123	<code>\num[minimum-integer-digits=4]{123}</code>
0.123	<code>\num{0.123}</code>
0.123	<code>\num[minimum-decimal-digits=2]{0.123}</code>
0.1230	<code>\num[minimum-decimal-digits=4]{0.123}</code>

### 4.5 打印数字

实际上，打印数字 (printing numbers) 是由许多设置控制的，这些设置应用了不同的小数点标记 (decimal markers)、数字分组 (digit grouping) 等思想。所有这些选项都与

输出的外观 (appearance of output) 有关, 而不是与输出所传递的数据 (data it conveys) 有关。表 13 总结了这些选项。

表 13: 数字 (numbers) 的输出选项

选项的名称	含义	类型	默认
bracket-negative-numbers	括号负数	Switch	false
digit-group-size	位数组大小	Integer	3
digit-group-first-size	第一数字组大小	Integer	3
digit-group-other-size	其它数字组大小	Integer	3
exponent-base	指数基数	Literal	10
exponent-product	指数乘积	Math	\times
group-digits	分组数字	Choice	all
group-minimum-digits	组最小位数	Integer	5
group-separator	组分隔符	Literal	\,
negative-color	负色	Literal	<i>none</i>
output-close-uncertainty	输出关不确定性	Literal	)
output-decimal-marker	输出小数点标记	Literal	.
output-exponent-marker	输出指数标记	Literal	<i>none</i>
output-open-uncertainty	输出开不确定性	Literal	(
print-implicit-plus	打印隐式加号	Switch	false
print-unity-mantissa	打印单位尾数	Switch	true
print-zero-exponent	打印零指数	Switch	false
print-zero-integer	打印零整数	Switch	true
tight-spacing	紧凑的间距	Switch	false
uncertainty-descriptor-mode	不确定性描述符模式	Choice	bracket-separator
uncertainty-descriptor-separator	不确定性描述分隔符	Literal	\
uncertainty-descriptors	不确定性描述符	Literal	<i>none</i>
uncertainty-mode	不确定性模式	Choice	compact
uncertainty-separator	不确定性分隔符	Literal	<i>none</i>
zero-decimal-as-symbol	零小数符号	Switch	false
zero-symbol	零符号	Literal	\mbox{---}

将数字分组为三个块 (blocks) 是一种常见的方法, 可以提高数字的易读性。  
`group-digits (env.) group-digits` (组数字) 选项控制此行为是否适用, 其取值可以是 `all`、`none`、`decimal`、`integer`。可以使用适当命名的值 (appropriately-named values) 分别为数字的整数和小数部分激活分组。

排版的结果	输入的命令
12 345.678 90	<code>\num{12345.67890}</code>
12345.67890	<code>\num[group-digits=none]{12345.67890}</code>
12345.678 90	<code>\num[group-digits=decimal]{12345.67890}</code>
12 345.67890	<code>\num[group-digits=integer]{12345.67890}</code>

`group-separator (env.)` 数字组 (groups of digits) 之间使用的分隔符 (separator) 由 `group-separator` (组分隔) 符选项存储。它接受文字输入 (literal input), 可以在数学模式 (math mode) 中使用: 对于文本模式 (text-mode), 全空格 (full space) 使用 `\quad`。

排版的结果	输入的命令
12 345	<code>\num{12345}</code>
12,345	<code>\num[group-separator={,}]{12345}</code>
12 345	<code>\num[group-separator=\ ]{12345}</code>

分组 (grouping) 并不总是适用于较小的数字 (smaller numbers); 这可以使用选项 `group-minimum-digits (env.) group-minimum-digits` (组最小位数) 来控制, 该选项指定在应用分组之前必须存在多少个数字。数字的整数部分和小数部分是分开考虑的: 分组不会“跨越边界 (cross the boundary)”。

排版的结果	输入的命令
1234	<code>\num{1234}</code>
12 345	<code>\num{12345}</code>
1234	<code>\num[group-minimum-digits=5]{1234}</code>
12 345	<code>\num[group-minimum-digits=5]{12345}</code>
1234.5678	<code>\num{1234.5678}</code>
12 345.678 90	<code>\num{12345.67890}</code>
1234.5678	<code>\num[group-minimum-digits=5]{1234.5678}</code>
12 345.678 90	<code>\num[group-minimum-digits=5]{12345.67890}</code>

`digit-group-size` (*env.*) 每组中的位数可以通过设置 `digit-group-size` (位数组大小) 来控制, 其标准值是 `digit-group-first-size` 3。使用 `digit-group-first-size` (第一数字组大小) 和 `digit-group-other-size` (其他数字组大小) 可以实现更精细的控制: 第一组是小数点 (decimal point) 之后的值, 其他值应用于第二组和后续组。例如, 这些可以用于实现印度 (lakh) 通常使用的分组。

排版的结果	输入的命令
1 234 567 890	<code>\num{1234567890}</code>
12345 67890	<code>\num[digit-group-size=5]{1234567890}</code>
1 23 45 67 890	<code>\num[digit-group-other-size=2]{1234567890}</code>

`output-decimal-marker` (*env.*) 输出中使用的小数点标记 (decimal marker) 是使用 `output-decimal-marker` (输出小数点标记) 选项来设置的; 这可能与输入标记 (input marker) 不同。

排版的结果	输入的命令
1.23	<code>\num{1.23}</code>
1,23	<code>\num[output-decimal-marker={,}]{1.23}</code>

`exponent-base` (*env.*) 当输入中出现指数 (exponents) 时, `exponent-base` (指数基) 选项和 `exponent-product` (指数乘积) 选项设置输出的明显部分 (obvious parts)。

排版的结果	输入的命令
$1 \times 10^2$	<code>\num[exponent-product=\times]{1e2}</code>
$1 \cdot 10^2$	<code>\num[exponent-product=\cdot]{1e2}</code>
$1 \times 2^2$	<code>\num[exponent-base=2]{1e2}</code>

`output-exponent-marker` 或者, 如果设置了 `output-exponent-marker` (输出指数标记) 选项, 则将使用存储的值 (*env.*) 代替正常的乘积 (product) 和基数 (base) 的组合 (combination)。

排版的结果	输入的命令
1e2	<code>\num[output-exponent-marker=e]{1e2}</code>
1E2	<code>\num[output-exponent-marker=\mathrm{E}]{1e2}</code>

当输入包含单个不确定性 (single uncertainty) 时, 可以将不确定性放在括号中或作为独立的数字 (separate number) 打印。这种行为受 `uncertainty-mode` (不确定性模式) 选

择的控制。当将其设置为 `separate` (单独) 时, 不确定性将打印为一个完全独立的数字 (entirely separate number), 前面加上 `\pm`。其他设置都将不确定性放在直接附加到主  
值 (main value) 的括号中。`compact` (紧凑的) 标准设置 (standard setting) 以最低有效位  
数 (least-significant digits) 打印不确定性数字。如果不确定性 (uncertainty) 超过了小数  
点 (crosses the decimal), 则不打印小数点标记 (decimal marker)。设置 `full` 打印不确  
定性的全部值 (full value)。最后, `compact-marker` (紧凑标记) 可以以 `compact` 样式打  
印, 除非不确定性超过了小数点, 在这种情况下使用 `full` 样式。当不确定性放在括号  
中时, 可以在主数字 (main number) 和不确定性之间添加一个空格 (space): 这是使用

`uncertainty-separator (env.)` `uncertainty-separator` (不确定性分隔符) 选项来存储的。所用的开括号 (opening) 存储  
`output-open-uncertainty` 在 `output-open-uncertainty` (输出开不确定性), 所用的闭括号 (closing brackets) 存储  
`output-close-uncertainty (env.)` 在 `output-close-uncertainty` (输出关不确定性) 中。可以使用 `uncertainty-separator`  
`(env.)` (不确定性分隔符) 将令牌 (tokens) 插入到开括号 (opening bracket) 之前。

排版的结果	输入的命令
123.45(120)	<code>\num{123.45(120)}</code>
0.035(14)	<code>\num{0.035(14)}</code>
123.45(1.20)	<code>\sisetup{uncertainty-mode=full}\num{123.45(120)}</code>
0.035(0.014)	<code>\sisetup{uncertainty-mode=full}\num{0.035(14)}</code>
123.45(1.20)	<code>\sisetup{uncertainty-mode=compact-marker}\num{123.45(120)}</code>
0.035(14)	<code>\sisetup{uncertainty-mode=compact-marker}\num{123.45(120)}</code>
1.234 ± 0.005	<code>\sisetup{uncertainty-mode=separate,output-open-uncertainty =[,output-close-uncertainty=],uncertainty-separator=\,} \num{1.234(5)}</code>

一个数字可以给出多种不确定性。这些值总是以独立的形式 (separated form) 打印。当  
存在多个不确定部分 (uncertainty part) 时, 描述该值的性质 (nature) 可能很有用。这可  
`uncertainty-descriptors` 以使用 `uncertainty-descriptors` (不确定性描述符) 选项来实现, 该选项采用逗号分隔  
`(env.)` 的描述列表。

`uncertainty-descriptor-separator` 可以使用 `uncertainty-descriptor-separator` (不确定性描述符分隔符) 来调  
`(env.)` 整描述符 (descriptors) 的格式, 描述符 (descriptors) 的格式也可以使用不确定性描述符  
`uncertainty-descriptor-mode` 模式即 `uncertainty-descriptor-mode` 来调整。`mode` (模式) 的选项有 `bracket` (括号)、  
`(env.)` `bracket-separator` (括号分隔符)、`separator` (分隔符)、`subscript` (下标)。

排版的结果	输入的命令
$1.2 \pm 0.3 \pm 0.4$	<code>\num{1.2(3)(4)}</code>
$1.2 \pm 0.3 \text{ (sys)} \pm 0.4 \text{ (stat)}$	<code>\sisetup{uncertainty-descriptors={sys,stat}}</code> <code>\num{1.2(3)(4)}</code>
$1.2 \pm 0.3_{\text{sys}} \pm 0.4_{\text{stat}}$	<code>\sisetup{uncertainty-descriptors={sys,stat}}</code> <code>\num[uncertainty-descriptor-mode=subscript]{1.2(3)(4)}</code>

数字输入 (numerical input) 的某些组合 (certain combinations) 可能不明确。这可以通过 `bracket-ambiguous-numbers` 在适当的位置添加括号 (brackets) 来纠正, 并由 `bracket-ambiguous-numbers` (括号不明 (env.) 确数字) 开关控制。此选项仅适用于纯数字 (pure numbers): 格式化数量 (quantities) 时, 括号的需要也取决于单位的位置, 因此由 `separate-uncertainty-units` (独立的不确定性单位) 控制。

排版的结果	输入的命令
$(1.2 \pm 0.3) \times 10^4$	<code>\sisetup{separate-uncertainty}\num{1.2(3)e4}</code>
$1.2 \pm 0.3 \times 10^4$	<code>\sisetup{separate-uncertainty}</code> <code>\num[bracket-ambiguous-numbers=false]{1.2(3)e4}</code>

`negative-color` (env.) 该宏包可以检测负尾数值 (negative mantissa values) 并相应地改变打印颜色 (print color)。通过将选项设置为空值, 可以禁用此选项。

排版的结果	输入的命令
$-15\,673$	<code>\num{-15673}</code>
$-15\,673$	<code>\num[negative-color=red]{-15673}</code>

在财务情况 (financial situations) 中显示负数 (negative numbers) 的常用方法是将它们放在括号中。这可以使用 `bracket-negative-numbers` (括号负数) 选项自动执行。

(env.)

排版的结果	输入的命令
$-15\,673$	<code>\num{-15673}</code>
$(15\,673)$	<code>\num[bracket-negative-numbers]{-15673}</code>
$-10\,\text{m}$	<code>\qty{-10}{\metre}</code>
$(10)\,\text{m}$	<code>\qty[bracket-negative-numbers]{-10}{\metre}</code>

`tight-spacing (env.)` 在某些情况下，可能需要“压缩 (squeeze)”输出间距 (output spacing)。这是使用 `tight-spacing` (紧凑的间距) 开关打开的，该开关在可能的情况下压缩间距。

排版的结果	输入的命令
$2 \times 10^3$	<code>\num{2e3}</code>
$2 \times 10^3$	<code>\num[tight-spacing=true]{2e3}</code>

`print-implicit-plus (env.)` 强制所有数字都有一个符号 (sign) 可能是有用的。这种行为由 `print-implicit-plus` (打印隐式加号) 选项控制：如果输入中没有符号，则使用该选项。

排版的结果	输入的命令
345	<code>\num{345}</code>
+345	<code>\num[print-implicit-plus]{345}</code>

尾数 (mantissa) 为 1、指数 (exponent) 为 0 和整数分量 (integer component) 为 0 的打印可通过选项 `print-unity-mantissa` (打印单位尾数)、`print-zero-exponent` (打印零指数)、`print-zero-integer` (打印零整数) 进行控制。标准设置打印尾数 1 和整数部分 (integer part) 0，但忽略指数 0。请注意，如果 `print-unity-mantissa` 和 `print-zero-exponent` 都设置为 `false`，则仍将打印值 1 (即 `print-zero-exponent` 具有更高的优先级)。

排版的结果	输入的命令
$1 \times 10^4$	<code>\num{1e4}</code>
$10^4$	<code>\num[print-unity-mantissa=false]{1e4}</code>
444	<code>\num{444e0}</code>
$444 \times 10^0$	<code>\num[print-zero-exponent=true]{444e0}</code>
0.123	<code>\num{0.123}</code>
.123	<code>\num[print-zero-integer=false]{0.123}</code>

在某些领域，特别是金融 (financial) 领域，完全零小数 (entirely zero decimal) 部分被破折号 (dash) 取代。这是由选项 `zero-decimal-as-symbol` (零小数符号) 支持的，`zero-symbol (env.)` `zero-decimal-as-symbol` 然后使用使用 `zero-symbol` (零符号) 存储的材料作为替换 (replacement)。

排版的结果	输入的命令
123.00	<code>\num{123.00}</code>
123.—	<code>\sisetup{zero-decimal-as-symbol}\num{123.00}</code>
123.[—]	<code>\sisetup{zero-decimal-as-symbol}</code> <code>\num[zero-symbol=\text{[---]}]{123.00}</code>

## 4.6 列表、乘积和范围

列表 (lists)、乘积 (products) 和数量范围 (ranges of numbers) 有少量专门的选项, 这些选项适用于这些不常见的输入形式 (unusual input forms)(表 14)。

表 14: 列表、乘积和数量范围 (ranges of numbers) 的输出选项

选项的名称	含义	类型	默认*
<code>list-exponents</code>	列表指数	Choice	<code>individual</code>
<code>list-final-separator</code>	列表最终分隔	Literal	<code>\text{and}</code>
<code>list-pair-separator</code>	列表成对分隔符	Literal	<code>\text{and}</code>
<code>list-separator</code>	列表分隔符	Literal	<code>\text{,}</code>
<code>list-units</code>	列表单位	Choice	<code>repeat</code>
<code>product-exponents</code>	乘积指数	Choice	<code>individual</code>
<code>product-mode</code>	乘积模式	Choice	<code>symbol</code>
<code>product-phrase</code>	乘积短语	Literal	<code>\text{by}</code>
<code>product-symbol</code>	乘积符号	Literal	<code>\times</code>
<code>product-units</code>	乘积单位	Choice	<code>repeat</code>
<code>range-exponents</code>	范围指数	Choice	<code>individual</code>
<code>range-phrase</code>	范围短语	Literal	<code>\text{to}</code>
<code>range-units</code>	范围单位	Choice	<code>repeat</code>

\* 默认值实际上更为复杂, 原因有二: 允许空格 (spaces) 在数学模式和文本模式中工作, 以及字符串 (strings) 的局部化 (localization)。

数字列表 (lists of numbers) 在每个条目 (item) 之间使用分隔符 (separator) 打印, 并 `list-separator (env.)` 使用 `list-separator` (分隔符列表) 选项存储。列表最后一项之前的分隔符可能不同, `list-final-separator (env.)` 因此使用 `list-final-separator` (列表最终分隔符) 选项进行设置。使用列表成对分隔符选项即 `list-pair-separator` 设置正好用于两个条目的分隔符。所需的任何空格

(spaces) 都应包含在选项设置中：代码中不添加任何空格。分隔符始终以文本模式 (text mode) 打印。

排版的结果	输入的命令
0.1, 0.2 and 0.3	<code>\numlist{0.1;0.2;0.3}</code>
0.1; 0.2 and 0.3	<code>\numlist[list-separator={; }]{0.1;0.2;0.3}</code>
0.1, 0.2, 0.3	<code>\numlist[list-final-separator={, }]{0.1;0.2;0.3}</code>
0.1 and 0.2 and finally 0.3	<code>\numlist[list-separator={ and },</code> <code>list-final-separator={ and finally }]</code> <code>{0.1;0.2;0.3}</code>
0.1 and 0.2	<code>\numlist{0.1;0.2}</code>
0.1, and 0.2	<code>\numlist[list-pair-separator={, and }]{0.1;0.2}</code>

数字 (numbers) 的乘积 (products) 可以使用乘积符号 (product symbol) 或短语 (phrase) `product-mode (env.)` 输出：这由 `product-mode` (乘积模式) 设置控制。设置 `symbol` (符号) 时，相应的符号 `product-symbol (env.)` 号存储在 `product-symbol` (乘积符号) 中。使用 `phrase-mode` (短语模式) 时，信息以 `product-phrase (env.)` `product-phrase` (乘积短语) 的形式存储。短语总是以文本模式 (text mode) 打印；符号打印使用与数字相同的程序。

排版的结果	输入的命令
$5 \times 100 \times 2$	<code>\numproduct{5 x 100 x 2}</code>
$5 \cdot 100 \cdot 2$	<code>\numproduct[product-symbol=\ensuremath{\cdot}]{5 x 100 x 2}</code>
5 by 100 by 2	<code>\sisetup{product-mode=phrase}\numproduct{5 x 100 x 2}</code>
5 BY 100 BY 2	<code>\sisetup{product-mode=phrase}\numproduct[product-phrase={ BY }]{5 x 100 x 2}</code>

数字的范围 (ranges of numbers) 可以作为输入。它们将在两个条目 (entries) 之间插入 `range-phrase (env.)` 适当的单词 (word) 或符号 (symbol)：这是使用 `range-phrase` (范围短语) 选项存储的。短语应包含任何必要的空格 (spaces)：不添加额外的空格。短语始终以文本模式 (text mode) 打印。

排版的结果	输入的命令
5 to 100	<code>\numrange{5}{100}</code>
5–100	<code>\numrange[range-phrase=--]{5}{100}</code>

通过组合指数部分 (combining the exponent parts), 可以“压缩 (compressed)”列  
list-exponents (env.) 表 (lists)、乘积 (products) 和范围 (ranges)。这是由 list-exponents (列表指数)、  
product-exponents (env.) product-exponents (乘积指数) 和 range-exponents (范围指数) 等选项控制的, 所  
range-exponents (env.) 有这些都采用 individual (单独)、combine-bracket (组合括号) 和 combine (组合) 选项。  
标准设置 individual 将指数 (exponent) 与匹配值 (matching value) 一起保留。combine  
和 combine-bracket 都取第一个条目的指数, 并应用于所有其他条目, 指数本身放在末  
尾 (end)。

排版的结果	输入的命令
$5 \times 10^3, 7 \times 10^3, 9 \times 10^3$ and $1 \times 10^4$	<code>\numlist{5e3;7e3;9e3;1e4}</code>
$5 \times 10^3 \times 7 \times 10^3 \times 9 \times 10^3 \times 1 \times 10^4$	<code>\numproduct{5e3 x 7e3 x 9e3 x 1e4}</code>
$5 \times 10^3$ to $7 \times 10^3$	<code>\numrange{5e3}{7e3}</code>
$(5, 7, 9$ and $10) \times 10^3$	<code>\sisetup{list-exponents=combine-bracket, product-exponents=combine-bracket, range-exponents=combine-bracket} \numlist{5e3;7e3;9e3;1e4}</code>
$(5 \times 7 \times 9 \times 10) \times 10^3$	<code>\sisetup{list-exponents=combine-bracket, product-exponents=combine-bracket, range-exponents=combine-bracket} \numproduct{5e3 x 7e3 x 9e3 x 1e4}</code>
$(5$ to $7) \times 10^3$	<code>\sisetup{list-exponents=combine-bracket, product-exponents=combine-bracket, range-exponents=combine-bracket} \numrange{5e3}{7e3}</code>
$5, 7, 9$ and $10 \times 10^3$	<code>\sisetup{list-exponents=combine, product-exponents=combine, range-exponents=combine} \numlist{5e3;7e3;9e3;1e4}</code>
$5 \times 7 \times 9 \times 10 \times 10^3$	<code>\sisetup{list-exponents=combine, product-exponents=combine, range-exponents=combine} \numproduct{5e3 x 7e3 x 9e3 x 1e4}</code>
$5$ to $7 \times 10^3$	<code>\sisetup{list-exponents=combine, product-exponents=combine, range-exponents=combine} \numrange{5e3}{7e3}</code>

`list-units (env.)` `list-units` (列表单位) 选项决定 `\qtylist` 命令打印单位 (units) 的方式, (乘积单位) `product-units (env.)` 选项即 `product-units` 决定 `\qtyproduct` 命令打印单位 (units) 的方式, 而 (范围单位) `range-units (env.)` 选项即 `range-units` 决定 `\qtyrange` 命令打印单位 (units) 的方式。这些的标准设置是 `repeat` (重复的), 其中每个数字 (number) 都将打印一个单位 (unit)。备选方案有

bracket (括号) 和 single (单个)。如果设置为 single, 这将覆盖指数的集合 (collection of exponents)。

排版的结果	输入的命令
2 T, 4 T, 6 T and 8 T	<code>\qtylist{2;4;6;8}{\tesla}</code>
(2, 4, 6 and 8) T	<code>\qtylist[list-units=bracket]{2;4;6;8}{\tesla}</code>
2 T, 4 T, 6 T and 8 T	<code>\qtylist[list-units=repeat]{2;4;6;8}{\tesla}</code>
2, 4, 6 and 8 T	<code>\qtylist[list-units=single]{2;4;6;8}{\tesla}</code>
2 °C to 4 °C	<code>\qtyrange{2}{4}{\degreeCelsius}</code>
(2 to 4) °C	<code>\qtyrange[range-units=bracket]{2}{4}{\degreeCelsius}</code>
2 °C to 4 °C	<code>\qtyrange[range-units=repeat]{2}{4}{\degreeCelsius}</code>
2 to 4 °C	<code>\qtyrange[range-units=single]{2}{4}{\degreeCelsius}</code>

product-units (乘积单位) 选项还提供了 bracket-power (括号幂) 和 power (幂) 的设置。

排版的结果	输入的命令
$2\,\text{m} \times 4\,\text{m}$	<code>\qtyproduct{2 x 4}{\metre}</code>
$(2 \times 4)\,\text{m}^2$	<code>\qtyproduct[product-units=bracket-power]{2 x 4}{\metre}</code>
$2 \times 4\,\text{m}^2$	<code>\qtyproduct[product-units=power]{2 x 4}{\metre}</code>

### 4.7 复数

少数选项专门用于处理复数 (complex numbers), 这些选项总结在表 15 中。

表 15: 用于处理复数 (complex numbers) 的选项

选项的名称	含义	类型	默认
<code>complex-angle-unit</code>	复角单位	Choice	degrees
<code>complex-mode</code>	复数模式	Choice	input
<code>complex-root-position</code>	复根位置	Choice	after-number
<code>complex-symbol-angle</code>	复数符号角	Literal	<code>\angle</code>
<code>complex-symbol-degree</code>	复数符号度	Literal	<code>\degree</code>
<code>input-complex-root</code>	输入复根	Literal	<code>ij</code>
<code>output-complex-root</code>	输出复根	Literal	<code>\mathrm{i}</code>
<code>print-complex-unity</code>	打印复数单位	Switch	false

`complex-mode (env.)` 可以使用 `complex-mode` (复数模式) 选项设置打印复数值 (complex values) 的格式。使用标准设置 `input` (输入), 将按给定方式打印复数值。通过将选项设置为 `cartesian` (笛卡尔) 或 `polar` (极坐标), 可以将输出格式 (output format) 设置为笛卡尔格形式或极坐标形式。转换使用  $\text{\LaTeX}$ 3 浮点单位 (floating-point unit), 所以小数点后的位数限制为 16 位。当从笛卡尔坐标转换为极坐标时, 复数根符号 (complex root symbol) 必须位于虚部 (imaginary part) 的末尾。它还必须使用 `i` 指定。

排版的结果	输入的命令
$1 + i$	<code>\complexnum{1 + i}</code>
$1\angle 45^\circ$	<code>\complexnum{1:45}</code>
$1 + i$	<code>\complexnum[complex-mode=cartesian]{1 + i}</code>
$0.71 + 0.71i$	<code>\complexnum[complex-mode=cartesian,round-mode=places]{1:45}</code>
$1\angle 0^\circ$	<code>\complexnum[complex-mode=polar]{1 + i}</code>
$1\angle 45^\circ$	<code>\complexnum[complex-mode=polar]{1:45}</code>

在输入中使用复数 (complex numbers) 时, 复根 (complex root) ( $i = \sqrt{-1}$ ) 由存储在 `input-complex-root (env.)` `input-complex-roots` (输入复根) 中的一个令牌 (tokens) 表示。解析器 (parser) 理解在相关数字 (但将检测任何无效排列) 之前或之后给出的复根符号 (complex root symbols):

排版的结果	输入的命令
$9.99 + 88.8i$	<code>\complexnum{9.99 + 88.8i}</code>
$9.99 + 88.8i$	<code>\complexnum{9.99 + i88.8}</code>

`output-complex-root (env.)` 输出复根符号 (output complex root symbol) 与输入无关, 可以使用 `output-complex-root` (输出复根) 设置进行更改。

排版的结果	输入的命令
$1 + 2i$	<code>\complexnum[output-complex-root=i]{1+2i}</code>
$1 + 2j$	<code>\complexnum[output-complex-root=j]{1+2i}</code>

`complex-root-position (env.)` 可以使用 `complex-root-position` (复根位置) 选项调整复根的位置 (position of the complex root), 以将其放置在复数中的相关数字之前或之后。

排版的结果	输入的命令
$67 - 0.9i$	<code>\complexnum{67-0.9i}</code>
$67 - i0.9$	<code>\complexnum[complex-root-position=before-number]{67-0.9i}</code>
$67 - 0.9i$	<code>\complexnum[complex-root-position=after-number]{67-0.9i}</code>

`complex-angle-unit (env.)` 当打印或转换为极坐标形式 (polar form) 时, 角度 (angle) 可以用 `complex-angle-unit` (复角单位) 设置的单位 (units) 来解释: `degrees` (度) 或 `radians` (弧度) 之一。用于表示 `complex-symbol-angle (env.)` 角度 (angle) 的符号由选项 `complex-symbol-angle` (复数符号角) 控制, 用于表示度数 `complex-symbol-degree (env.)` (degrees) 单位的符号则由选项 `complex-symbol-degree` (复数符号度) 控制。

排版的结果	输入的命令
$1\angle 1^\circ \Omega$	<code>\complexqty{1:1}{\ohm}</code>
$1\angle 1 \Omega$	<code>\complexqty[complex-angle-unit=radians]{1:1}{\ohm}</code>
$1A1^\circ \Omega$	<code>\complexqty[complex-symbol-angle=\mathrm{A}]{1:1}{\ohm}</code>
$1\angle 1d \Omega$	<code>\complexqty[complex-symbol-degree=d]{1:1}{\ohm}</code>

当数字的复数部分 (complex part) 正好为 1 时, 可以打印或者禁止显示该值。这是由 `print-complex-unity (env.)` 关 `print-complex-unity` (打印复数单位) 控制的。

排版的结果	输入的命令
$i \Omega$	<code>\complexqty{1i}{\ohm}</code>
$1i \Omega$	<code>\complexqty[print-complex-unity]{1i}{\ohm}</code>

## 4.8 角度

由 `\ang` 函数提供的角度处理 (angle processing) 有一组选项, 这些选项除了用于数字处理的一般选项之外, 还适用于其他选项。

表 16: 角度 (angle) 选项

选项的名称	含义	类型	默认
<code>angle-mode</code>	角度模式	Choice	<code>input</code>
<code>angle-symbol-degree</code>	角度符号度	Literal	<code>\degree</code>
<code>angle-symbol-minute</code>	角度符号分	Literal	<code>\arcminute</code>
<code>angle-symbol-over-decimal</code>	小数点标记上的角度符号	Switch	<code>false</code>
<code>angle-symbol-second</code>	角度符号秒	Literal	<code>\arcsecond</code>
<code>angle-separator</code>	角度分隔符	Literal	<code>\langle empty \rangle</code>
<code>fill-angle-degrees</code>	填充角度度	Switch	<code>false</code>
<code>fill-angle-minutes</code>	填充角度分	Switch	<code>false</code>
<code>fill-angle-seconds</code>	填充角度秒	Switch	<code>false</code>
<code>number-angle-product</code>	数字角乘积	Literal	<code>\langle empty \rangle</code>

`angle-mode (env.)` 可以使用 `angle-mode` (角度模式) 选项设置打印角度的格式。使用标准设置 `input` (输入), 可以按照给定的方式打印角度。通过将该选项设置为 `arc` (弧) 或 `decimal` (小数), 输出格式可以设置为弧 (度/分/秒) 或小数值 (decimal value)。转换使用 L<sup>A</sup>T<sub>E</sub>X3 浮点单位 (floating-point unit), 所以小数点后的位数限制为 16 位。

排版的结果	输入的命令
$2.67^\circ$	<code>\ang{2.67}</code>
$2^\circ 3' 4''$	<code>\ang{2;3;4}</code>
$2^\circ 40' 12''$	<code>\ang[angle-mode=arc]{2.67}</code>
$2^\circ 3' 4''$	<code>\ang[angle-mode=arc]{2;3;4}</code>
$2.67^\circ$	<code>\ang[angle-mode=decimal]{2.67}</code>
$2.051\,111\,111\,111\,111^\circ$	<code>\ang[angle-mode=decimal]{2;3;4}</code>

数字 (number) 和角度符号 (angle symbol)(度、分或秒) 之间的分隔符 (separator) 可以使用 `number-angle-product (env.)` 以使用 `number-angle-product` (数字角乘积) 选项来设置, 与 `\qty` 命令使用的相关 `quantity-product` (数量乘积) 选项无关。

排版的结果	输入的命令
$2.67^\circ$	<code>\ang{2.67}</code>
$2.67^\circ$	<code>\ang[number-angle-product=\,]{2.67}</code>

`angle-separator (env.)` 当角度 (angles) 以圆弧格式 (arc format) 打印时, 使用 `arc-separator` (弧分隔符) 选项设置不同部分的分隔。

排版的结果	输入的命令
$6^{\circ}7'6.5''$	<code>\ang{6;7;6.5}</code>
$6^{\circ} 7' 6.5''$	<code>\ang[angle-separator=\,]{6;7;6.5}</code>

`fill-angle-degrees (env.)` 弧 (arc) 的度 (degree) 的零填充 (zero-filling) 由 `fill-angle-degrees` (填充角度度) 选项控制, 弧 (arc) 的分 (minute) 的零填充 (zero-filling) 由 `fill-angle-minutes` (填充角度分) 选项控制, 弧 (arc) 的秒 (second) 的零填充 (zero-filling) 由 `fill-angle-seconds` (填充角度秒) 选项控制。所有的都是标准配置。

排版的结果	输入的命令
$-1^{\circ}$	<code>\ang{-1;;}</code>
$-2'$	<code>\ang{;-2;}</code>
$-3''$	<code>\ang{;;-3}</code>
$-1^{\circ}$	<code>\sisetup{fill-angle-degrees}\ang{-1;;}</code>
$-0^{\circ}2'$	<code>\sisetup{fill-angle-degrees}\ang{;-2;}</code>
$-0^{\circ}3''$	<code>\sisetup{fill-angle-degrees}\ang{;;-3}</code>
$-1^{\circ}0'$	<code>\sisetup{fill-angle-minutes}\ang{-1;;}</code>
$-2'$	<code>\sisetup{fill-angle-minutes}\ang{;-2;}</code>
$-0'3''$	<code>\sisetup{fill-angle-minutes}\ang{;;-3}</code>
$-1^{\circ}0''$	<code>\sisetup{fill-angle-seconds}\ang{-1;;}</code>
$-2'0''$	<code>\sisetup{fill-angle-seconds}\ang{;-2;}</code>
$-3''$	<code>\sisetup{fill-angle-seconds}\ang{;;-3}</code>

在标准设置 (standard settings) 中, 用于弧角 (arc angles) 的符号是单位命令 (`unit angle-symbol-degree (env.) commands`) `\degree`、`\arcminute` 和 `\arcsecond`。可以使用 `angle-symbol-degree` (角度符号度) 选项来更改 `\degree` 的值, 可以使用 `angle-symbol-minute` (角度符号分) 选项来更改 `\arcminute` 的值, 可以使用 `angle-symbol-second` (角度符号秒) 选项来更改 `\arcsecond` 的值。当更改单位宏 (unit macros) 的定义时, 例如将 `\arcsecond` 设置为 `as` 时, 最可能使用此选项。

排版的结果	输入的命令
$6^{\circ}7'6.5''$	<code>\ang{6;7;6.5}</code>
$6\text{d}7\text{m}6.5\text{s}$	<code>\sisetup{angle-symbol-degree=d, angle-symbol-minute=m, angle-symbol-second=s} \ang{6;7;6.5}</code>

在一些学科领域, 尤其是天文学领域, 角度符号 (angle symbols) 是在小数点标记 (decimal angle-symbol-over-decimal marker) 上给出的, 而不是在数字的末尾。使用 `angle-symbol-over-decimal` (小数点标 (*env.*) 记上的角度符号) 选项可以实现此行为。

排版的结果	输入的命令
$45.697^{\circ}$	<code>\ang{45.697}</code>
$6^{\circ}7'6.5''$	<code>\ang{6;7;6.5}</code>
$45^{\circ}697$	<code>\ang[angle-symbol-over-decimal]{45.697}</code>
$6^{\circ}7'6.''5$	<code>\ang[angle-symbol-over-decimal]{6;7;6.5}</code>

## 4.9 创建单位

在文档的开头创建各种宏单元 (macro units)。siunitx 可以定义这些宏, 以便它们只能在 `\unit` 和 `\qty` 函数中使用, 或者可以在整个文档正文 (document body) 中使用单元宏 (unit macros)。有许多设置控制这个创建过程 (表 17)。因此, 这些选项都只适用于前言 (preamble) 部分。

表 17: 单位创建 (unit creation) 选项

选项的名称	含义	类型	默认
<code>free-standing-units</code>	独立单位	Switch	false
<code>overwrite-command</code>	覆盖命令	Switch	false
<code>space-before-unit</code>	单位前空格	Switch	false
<code>unit-optional-argument</code>	单位可选参数	Switch	false
<code>use-xspace</code>	使用 <code>xspace</code>	Switch	false

`free-standing-units (env.)` `free-standing-units` (独立单位) 选项控制单位宏 (unit macros) 是否存在于 `\unit` 和 `\qty` 参数之外。如果此选项为 `true`, `siunitx` 将创建宏以供一般使用 (general use)。实现此目的的标准方法不覆盖 (overwrite) 任何现有宏：可以使用 `overwrite-commands` (覆盖命令) 开关更改此行为。

当使用“独立 (free-standing)”单位命令 (unit commands) 的方法时，通常只定义使用 `\DeclareSIUnit` 创建的宏。因此，词头 (prefixes) 和幂 (powers) 应该与所需的单位组合成一个独立的命令 (a single free-standing command)，例如

```
\DeclareSIUnit\kilometre{\kilo\metre}
```

当“独立 (free standing)”单位宏 (unit macros) 被创建时，它们的行为可以通过许多选项来调整。这些主要用于模拟旧宏包 (older packages) 的输入语法 (input syntax)。选项

`unit-optional-argument` `unit-optional-argument` (单位可选参数) 为输入

```
(env.) \qty{10}{\metre}
```

和

```
\metre[10].
```

`space-before-unit (env.)` 提供相同的行为。`space-before-unit` (单位前空格) 选项和 `use-xspace` (使用 `xspace`) 选项控制单位宏 (unit macros) “末端 (ends)” 的行为。激活 `space-before-unit` (单位前空格) 插入数字 (number)–单位前的单位空格 (unit space) 将被打印。这适用于输入语法 (input syntax)

```
30\metre
```

但这意味着单位宏 (unit macros) 在运行文本 (running text) 中的空格不正确。另一方面，`use-xspace` (使用 `xspace`) 选项尝试正确地分隔输入，例如：

```
\metre is the symbol for metres.
```

## 4.10 使用单位

`Siunitx` 的部分功能是能够在不改变输入的情况下改变单位的输出格式。因此，单位的行为 (behaviour of units) 是由一些选项控制，这些选项改变单位的处理 (processing) 或直接输出 (表 18)。

表 18: 单位输出选项 (unit output options)

选项的名称	含义	类型	默认
<code>bracket-unit-denominator</code>	括号单位分母	Switch	<code>true</code>
<code>forbid-literal-units</code>	禁止文字单位	Switch	<code>false</code>
<code>fraction-command</code>	分数命令	Literal	<code>\frac</code>
<code>inter-unit-product</code>	单位间乘积	Literal	<code>\,</code>
<code>parse-units</code>	解析单位	Switch	<code>true</code>
<code>per-mode</code>	每模式	Choice	<code>power</code>
<code>per-symbol-script-correction</code>	每符号脚本校正	Literal	<code>\!</code>
<code>per-symbol</code>	每符号	Literal	<code>/</code>
<code>power-half-as-sqrt</code>	幂的一半作为平方根	Switch	<code>false</code>
<code>qualifier-mode</code>	限定符模式	Choice	<code>subscript</code>
<code>qualifier-phrase</code>	限定词短语	Literal	<code>\langle empty \rangle</code>
<code>sticky-per</code>	粘着每	Switch	<code>false</code>
<code>unit-font-command</code>	单位字体命令	Literal	<code>\mathrm</code>

`inter-unit-product` (*env.*) 每个单位之间的分隔符 (separator) 使用 `inter-unit-product` (单位间乘积) 选项进行存储。标准设置是一个很小的间距 (thin space): 另一个常见的选择是居中的点 (centred dot)。要获得正确的间距, 必须在后一种情况下使用 `\ensuremath{{}\cdot{}}`。

排版的结果	输入的命令
$F^2 \text{ lm cd}$	<code>\unit{\farad\squared\lumen\candela}</code>
$F^2 \cdot \text{ lm} \cdot \text{ cd}$	<code>\unit[inter-unit-product=\ensuremath{{}\cdot{}}]{\farad\squared\lumen\candela}</code>

`per-mode` (*env.*) 使用 `per-mode` (每模式) 选择选项更改 `\per` 的处理方式。标准设置是 `power` (幂), 这意味着 `\per` 为单位生成倒数幂 (reciprocal powers)。使用 `\frac` 函数将选项设置为 `fraction` (分数), 分别排版单位的正幂 (positive powers) 和负幂 (negative powers)。确切的函数

`fraction-command` (*env.*) (exact function) 可以使用 `fraction-command` (分数命令) 选项进行调整。

排版的结果	输入的命令
$\text{J mol}^{-1} \text{K}^{-1}$	<code>\unit{\joule\per\mole\per\kelvin}</code>
$\text{m s}^{-2}$	<code>\unit{\metre\per\second\squared}</code>
$\frac{\text{J}}{\text{mol K}}$	<code>\unit[per-mode=fraction]{\joule\per\mole\per\kelvin}</code>
$\frac{\text{m}}{\text{s}^2}$	<code>\unit[per-mode=fraction]{\metre\per\second\squared}</code>

密切相关的 `power-positive-first` (正幂优先) 设置以相同的方式起作用, 但将所有的正幂 (positive powers) 置于任何负幂 (negative ones) 之前。

排版的结果	输入的命令
$\text{A mol}^{-1} \text{s}$	<code>\unit{\ampere\per\mole\second}</code>
$\text{A s mol}^{-1}$	<code>\unit[per-mode=power-positive-first]{\ampere\per\mole\second}</code>

通过将 `per-mode` (每模式) 设置为 `symbol` (符号), 可以使用符号 (通常为/) 来分隔单位的 `per-symbol (env.)` 两个部分; 使用 `per-symbol` (每符号) 的设置来存储所使用的符号 (symbol)。这种显示 `bracket-unit-denominator` 单位的方法可能不明确 (ambiguous), 除非 `bracket-unit-denominator` (括号单位分母) (`env.`) 设置为 `false`, 否则会添加括号 (brackets)。请注意, `bracket-unit-denominator` (括号单位分母) 仅在 `per-mode` (每模式) 设置为 `symbol` (符号) 时适用。

排版的结果	输入的命令
$\text{J}/(\text{mol K})$	<code>\sisetup{per-mode=symbol}\unit{\joule\per\mole\per\kelvin}</code>
$\text{m}/\text{s}^2$	<code>\sisetup{per-mode=symbol}\unit{\metre\per\second\squared}</code>
$\text{J div}(\text{mol K})$	<code>\sisetup{per-mode=symbol}\unit[per-symbol=\ \text{div}\ ]{\joule\per\mole\per\kelvin}</code>
$\text{J}/\text{mol K}$	<code>\sisetup{per-mode=symbol}\unit[bracket-unit-denominator=false]{\joule\per\mole\per\kelvin}</code>

通常要求的 (但数学上无效的) `repeated-symbol` (重复符号) 选项也可用于重复每个 `\per` 符号。

排版的结果	输入的命令
$\text{J}/\text{mol}/\text{K}$	<code>\unit[per-mode=repeated-symbol]{\joule\per\mole\per\kelvin}</code>

一个符号 (symbol) 的使用可以被限制在恰好需要一个符号的情况下：当且仅当有一个或多个正幂 (positive powers) 和恰好一个负幂 (negative power) 时，设置 `single-symbol` (单个符号) 将使用一个符号。在其他情况下，使用幂 (powers)。

排版的结果	输入的命令
$10\,\mathrm{m}^{-1}$	<code>\sisetup{per-mode=single-symbol}</code> <code>\qty{10}{\per\metre}</code>
$20\,\mathrm{m/s}$	<code>\sisetup{per-mode=single-symbol}</code> <code>\qty{20}{\metre\per\second}</code>
$30\,\mathrm{J\,mol^{-1}\,K^{-1}}$	<code>\sisetup{per-mode=single-symbol}</code> <code>\qty{30}{\joule\per\mole\per\kelvin}</code>

`display-per-mode (env.)`  
`inline-per-mode (env.)`

`\per` 函数 (function) 的行为可能取决于流行的数学风格 (prevailing math style)。可以使用 `display-per-mode` (显示每模式) 或 `inline-per-mode` (行内每模式) 独立设置来实现这一点。例如，下面的示例将使用行内数学 (line math) 的 `symbol` (符号) 设置，以及显示数学 (display math) 中使用的 `fraction` (分数) 设置。

排版的结果	输入的命令
$\mathrm{J/(mol\,K)}$	<code>\sisetup{display-per-mode=fraction,inline-per-mode=symbol}</code> <code>\$ \unit{\joule\per\mole\per\kelvin} \$</code>
$\mathrm{J/(mol\,K)}$	<code>\sisetup{display-per-mode=fraction,inline-per-mode=symbol}</code> <code>\[ \unit{\joule\per\mole\per\kelvin} \]</code>
$\mathrm{J/(mol\,K)}$	<code>\sisetup{display-per-mode=fraction,inline-per-mode=symbol}</code> <code>\unit{\joule\per\mole\per\kelvin}</code>
$\frac{\mathrm{J}}{\mathrm{mol\,K}}$	<code>\sisetup{display-per-mode=fraction,inline-per-mode=symbol}</code> <code>\$ \displaystyle \unit{\joule\per\mole\per\kelvin} \$</code>
$\mathrm{J/(mol\,K)}$	<code>\sisetup{display-per-mode=fraction,inline-per-mode=symbol}</code> <code>\[ \textstyle\unit{\joule\per\mole\per\kelvin} \]</code>

`per-symbol-script-correction`

使用 `per-mode` (每模式) 的 `symbol` (符号) 设置时，可能需要调整上标幂 (superscript (env.) power) 和符号 (symbol) 之间的间距。这是作为要插入到两个条目之间的命令提供的：标准值是一个很小的负空格 (negative space) 即 `\!`。

排版的结果	输入的命令
$\text{cm}^3/\text{g}$	<code>\sisetup{per-mode=symbol}\unit{\cm\cubed\per\gram}</code>
$\text{cm}^3/\text{g}$	<code>\sisetup{per-mode=symbol}\unit[per-symbol-script-correction= ]{\cm\cubed\per\gram}</code>

`sticky-per (env.)` 默认情况下, `\per` 仅适用于给定的下一个单位<sup>4</sup>。通过设置 `sticky-per` (粘着每) 标志 (flag), 此行为将被更改, 从而 `\per` 适用于所有后续单位 (subsequent units)。

排版的结果	输入的命令
$\text{Pa Gy}^{-1} \text{H}$	<code>\unit{\pascal\per\gray\henry}</code>
$\text{Pa Gy}^{-1} \text{H}^{-1}$	<code>\unit[sticky-per]{\pascal\per\gray\henry}</code>

`qualifier-mode (env.)` 单位限定符 (unit qualifiers) 可以打印成三种不同的格式, 由 `qualifier-mode` (限定符模式) 选项设置。标准设置是 `subscript` (下标), 而选项 `bracket` (括号)、`combine` (组合) 和 `phrase` (短语) 也是可能的。在最后一个设置中, 幂 (powers) 可能导致歧义, 并且会被自动检测到并适当地添加括号 (brackets)。

排版的结果	输入的命令
$\text{kg}_{\text{pol}}^2 \text{mol}_{\text{cat}}^{-1} \text{h}^{-1}$	<code>\unit{\kilogram\of{pol}\squared\per\mole\of{cat}\per\hour}</code>
$\text{kg}(\text{pol})^2 \text{mol}(\text{cat})^{-1} \text{h}^{-1}$	<code>\unit[qualifier-mode=bracket]{\kilogram\of{pol}\squared\per\mole\of{cat}\per\hour}</code>
$\text{dBi}$	<code>\unit[qualifier-mode=combine]{\deci\bel\of{i}}</code>

`qualifier-phrase (env.)` `phrase` (短语) 选项与 `qualifier-phrase` (限定词短语) 一起使用, 例如允许插入空格 (space) 或其他链接文本 (linking text)。

<sup>4</sup>这是英语中阅读单位的标准方法: 例如,  $\text{J mol}^{-1} \text{K}^{-1}$  读作 “joules per mole per kelvin”。

排版的结果	输入的命令
$\text{kg pol}^2 \text{ mol cat}^{-1} \text{ h}^{-1}$	<code>\sisetup{qualifier-mode=phrase, qualifier-phrase=\ } \unit{\kilogram\of{pol} \squared\per\mole\of{cat}\per\hour}</code>
$\text{kg of pol}^2 \text{ mol of cat}^{-1} \text{ h}^{-1}$	<code>\sisetup{qualifier-mode=phrase, qualifier-phrase=\ \mbox{of}\ } \unit{\kilogram\of{pol} \squared\per\mole\of{cat}\per\hour}</code>

在某些情况下，通过将单位符号 (unit symbol) 作为平方根 (square root) 来表示 0.5 `power-half-as-sqrt` (*env.*) 的幂。这可以通过将 `power-half-as-sqrt` (幂的一半作为平方根) 设置为 `true` 来实现。

排版的结果	输入的命令
$\text{Hz}^{0.5}$	<code>\unit{\Hz\tothe{0.5}}</code>
$\sqrt{\text{Hz}}$	<code>\unit[power-half-as-sqrt]{\Hz\tothe{0.5}}</code>

通常，`siunitx` 在启用单位解析 (unit parse) 的情况下使用，只有在有文字输入 (literal input) 的情况下才直接打印单位。然而，如果已知输入基本上是一致的，并且希望获得高性能 (high performance)，那么可以使用 `parse-units` (解析单位) 开关关闭解析器 (parser)。

排版的结果	输入的命令
$300 \text{ MHz}$	<code>\qty{300}{\MHz}</code>
$300 \text{ MHz}$	<code>\qty[parse-units=false]{300}{\MHz}</code>

一些用户可能更倾向于完全禁用以单位表示的文字输入 (literal input)，例如为了强制 `forbid-literal-units` (*env.*) 一致性 (consistency)。这可以通过设置 `forbid-literal-units` (禁止文字单位) 开关来实现。启用此选项后，文档中只能使用基于宏的单位 (macro-based units)。这仅适用于 `parse-units` (解析单位) 为 `true` 时。

`unit-font-command` (*env.*) 可使用 `unit-font-command` (单位字体命令) 选项调整用于设置单位本身的命令。这通常设置为 `\mathrm`。

排版的结果	输入的命令
$\text{lm}$	<code>\unit{\lumen}</code>
$\textit{lm}$	<code>\unit[unit-font-command=\mathit]{\lumen}</code>

## 4.11 数量

一些选项适用于数量 (quantities) 即数字 (number) 和单位 (unit) 的组合，而不是仅适用于数字或单位 (表 19)。

表 19: 数量的选项 (options for quantities)

选项的名称	含义	类型	默认
<code>allow-quantity-breaks</code>	允许数量断开	Switch	false
<code>extract-mass-in-kilograms</code>	以 kg 为单位提取质量	Switch	true
<code>prefix-mode</code>	词头模式	Choice	input
<code>quantity-product</code>	数量乘积	Literal	$\cdot$
<code>separate-uncertainty-units</code>	独立的不确定性单位	Choice	bracket

通常，数字 (number) 和单位 (unit) 的组合 (combination) 被视为一个单独的数学实体 (single mathematical entity)，不应跨行拆分 (split across lines)。然而，在某些情况下 `allow-quantity-breaks` (env.) (非常长的单位、窄的列等) 可能需要中断。这可以使用 `allow-quantity-breaks` (允许数量断开) 选项打开。

Some filler text	<code>\begin{minipage}{2.55cm}</code>
10 m	<code>% Gives an underfull hbox</code>
Some filler text	<code>Some filler text \qty{10}{\metre} \\</code>
10 m	<code>\sisetup{allow-quantity-breaks}</code>
	<code>Some filler text \qty{10}{\metre}</code>
	<code>\end{minipage}</code>

`quantity-product` (env.) 数字 (number) 和单位 (unit) 之间的乘积符号 (product symbol) 使用 `quantity-product` (数量乘积) 选项设置。

排版的结果	输入的命令
2.67 F	<code>\qty{2.67}{\farad}</code>
2.67 F	<code>\qty[quantity-product=\ ]{2.67}{\farad}</code>
2.67F	<code>\qty[quantity-product=]{2.67}{\farad}</code>

单位词头 (unit prefixes)(\kilo等) 通常以字母 (letters) 形式给出。然而, 该宏包可以将这 `prefix-mode (env.)` 些转换为数字幂 (numerical powers)。这由 `prefix-mode` (词头模式) 选项控制, 该选项接受的值可以是 `input` (输入)、`combine-exponent` (组合的指数) 和 `extract-exponent` (提取的指数)。`input` (输入) 设置保持词头不变。使用 `combine-exponent` (组合的指数) 将把任何指数量 (any exponent amount) 从数字 (numbe) 添加到第一个单位 (first unit): 这将修改任何现有的词头。最后, 使用 `extract-exponent` (提取的指数) 将删除所有词头并将其表示为指数。在这种情况下, 千克 (kilograms) 的处理可以使用 `extract-mass-in-kilograms` `extract-mass-in-kilograms` (以 kg 为单位提取质量) 来设置: 如果为 `true`, `kilo` 词头 (`env.`) 将作为单位的一部分保留。这意味着所有克 (grams) 都换算成千克 (kilograms)。

排版的结果	输入的命令
$1 \times 10^3 \text{ ms}$	<code>\qty{1e3}{\metre\second}</code>
1 kms	<code>\qty[prefix-mode=combine-exponent]{1e3}{\metre\second}</code>
$10 \text{ kg}^2 \text{ ds}$	<code>\qty{10}{\kilo\gram\squared\deci\second}</code>
$10 \times 10^{-1} \text{ kg}^2 \text{ s}$	<code>\qty[prefix-mode=extract-exponent]{10}</code> <code>{\kilo\gram\squared\deci\second}</code>
$7.5 \times 10^{-3} \text{ kg}$	<code>\qty[prefix-mode=extract-exponent]{7.5}{\gram}</code>
$10 \text{ kg}^2 \text{ ds}$	<code>\sisetup{extract-mass-in-kilograms=false}</code> <code>\qty{10}{\kilo\gram\squared\deci\second}</code>
$10 \times 10^5 \text{ g}^2 \text{ s}$	<code>\sisetup{extract-mass-in-kilograms=false}</code> <code>\qty[prefix-mode=extract-exponent]{10}</code> <code>{\kilo\gram\squared\deci\second}</code>
7.5 g	<code>\sisetup{extract-mass-in-kilograms=false}</code> <code>\qty[prefix-mode=extract-exponent]{7.5}{\gram}</code>

当一个数字 (number) 有多个部分 (例如一个独立的不确定性 [`separate uncertainty`]) 时, `separate-uncertainty-units` 该单位必须适用于该数字的所有部分。使用 `separate-uncertainty-units` (独立的不确定性单位) 选项控制显示方式。标准设置为 `brackets` (括号), 将整个数字部分 (entire numerical part) 放在括号中, 并使用单个单位符号 (single unit symbol)。可选选项是 (`env.`)

repeat (重复)(为数字的每个部分打印单位) 和 single (单个)(仅打印一个单位符号: 数学错误 [mathematically incorrect])。

排版的结果	输入的命令
(12.3 ± 0.4) kg	<code>\sisetup{separate-uncertainty}</code> <code>\qty{12.3(4)}{\kilo\gram}</code>
(12.3 ± 0.4) kg	<code>\sisetup{separate-uncertainty}</code> <code>\qty[separate-uncertainty-units=bracket]{12.3(4)}{\kilo\gram}</code>
12.3 kg ± 0.4 kg	<code>\sisetup{separate-uncertainty}</code> <code>\qty[separate-uncertainty-units=repeat]{12.3(4)}{\kilo\gram}</code>
12.3 ± 0.4 kg	<code>\sisetup{separate-uncertainty}</code> <code>\qty[separate-uncertainty-units=single]{12.3(4)}{\kilo\gram}</code>

4.12 表格的材料

表中材料的处理遵循与已经描述的函数 (functions) 相同的设置。然而，有一些设置只适用于表格材料的布局 (表 20)。

`table-alignment-mode (env.)`      siunitx 用于对齐数字 (align numbers) 的方法是使用 `table-alignment-mode` (表格对齐模式) 选项来选择的，该选项可以是 `marker` (标记)、`format` (格式化) 或 `none` (无) 之一。使用标准设置 `marker` (标记)，宏包将小数点标记 (decimal marker) 在表格列居中，可能会在数字的短端 (shorter end) 留下空白 (space)。`format` (格式化) 模式 (mode) 使用 `table-format` (表格格式) 键中的信息来构造模型：然后使用它来定义数字的可用空间。对于非对称数 (asymmetrical numbers)，此方法非常可取。最后，`none` (无) 完全禁用对齐：只解析数字。

当 `table-alignment-mode` (表格对齐模式) 设置为 `format` (格式化) 或 `none` (无) 时，`table-number-alignment` 数字“块 (block)”在整个单元格中的位置由 `table-number-alignment` (表格数字对齐) 选项设置，该选项可以是 `left` (左)、`center` (中) 或 `right` (右)。当 `table-alignment-mode` (表格对齐模式) 设置为 `marker` (标记) 时，小数标记 (decimal marker) 始终居中于单元格。不同的对齐选择如表 21 所示，表 21 使用了一些夸张的列表头 (exaggerated column headings) 来显示单元格内容的相对位置。

表 20: 用于表格材料 (tabular material) 的选项

选项的名称	含义	类型	默认
<code>table-align-comparator</code>	表格对齐比较符	Switch	true
<code>table-align-exponent</code>	表格对齐指数	Switch	true
<code>table-align-text-after</code>	在后面对齐表格文本	Switch	true
<code>table-align-text-before</code>	在前面对齐表格文本	Switch	true
<code>table-align-uncertainty</code>	表格对齐不确定性	Switch	true
<code>table-alignment</code>	表格对齐	Meta	center
<code>table-alignment-mode</code>	表格对齐模式	Choice	marker
<code>table-auto-round</code>	表格自动舍入	Switch	false
<code>table-column-width</code>	表格列宽	Length	0pt
<code>table-fixed-width</code>	表格固定宽度	Switch	false
<code>table-format</code>	表格格式	Special	2.2
<code>table-number-alignment</code>	表格数字对齐	Choice	center
<code>table-text-alignment</code>	表格文本对齐	Choice	center

```

\begin{table}[H]
  \caption{\ {\Heiti 对齐} \texttt{S} {\Heiti 列}\label{tab:S:align}}
  \centering
  \sisetup{table-format = 2.4, table-alignment-mode = format}
  \begin{tabular}{@{}
    S[table-alignment-mode = marker]
    S[table-number-alignment = center]
    S[table-number-alignment = left]
    S[table-number-alignment = right]
  @{}}
  \hline{1.2pt}
  {\Heiti 一些值} & {\Heiti 一些值} & {\Heiti 一些值} & {\Heiti 一些值} \\
  \hline{0.7pt}
    2.3456 & 2.3456 & 2.3456 & 2.3456 \\
    34.2345 & 34.2345 & 34.2345 & 34.2345 \\
    56.7835 & 56.7835 & 56.7835 & 56.7835 \\
    90.473 & 90.473 & 90.473 & 90.473 \\
  \hline{1.2pt}

```

```
\end{tabular}  
\end{table}
```

表 21: 对齐 s 列

一些值	一些值	一些值	一些值
2.3456	2.3456	2.3456	2.3456
34.2345	34.2345	34.2345	34.2345
56.7835	56.7835	56.7835	56.7835
90.473	90.473	90.473	90.473

当对齐模式 (alignment mode) 设置为 none (无) 时, 只需收集和解析数字, 无需任何进一步处理, 如表 22 所示。

```
\begin{table}[H]  
  \caption{\ {\Heiti 在} \texttt{S} {\Heiti 列中只解析而不对齐}\label{tab:S:parse}}  
  \begin{tabular}  
    {@{  
      S  
      S[table-alignment-mode = none]  
    @{}}  
    \hline{1.2pt}  
    {\Heiti 小数点对齐} &  
    {\Heiti 简单居中} \\  
    \hline{0.7pt}  
    12.345 & 12.345 & \\  
    6,78 & 6,78 & \\  
    -88.8(9) & -88.8(9) & \\  
    4.5e3 & 4.5e3 & \\  
    \hline{1.2pt}  
  \end{tabular}  
\end{table}
```

表 22: 在 s 列中只解析而不对齐

小数点对齐	简单居中
12.345	12.345
6.78	6.78
-88.8(9)	-88.8(9)
$4.5 \times 10^3$	$4.5 \times 10^3$

当 `table-alignment-mode` (表格对齐模式) 设置为 `format` (格式化) 时, `siunitx` 使用 `table-format (env.)` 在 `table-format` (表格格式) 中设置的信息来构造一个“模型 (model)”, 该模型定义了为数字保留的空白 (space)。`table-format` (表格格式) 键的解释方式与表格单元格 (table cell) 大致相同。数字部分应包括一个数字, 显示输入的每个部分中要保留多少数字 (figures), 还有任何比较符 (comparators)、符号 (signs) 等。表 23 给出了各种示例。

```
\begin{table}[H]
\caption{\ {\Heiti 在}~\texttt{S}~{\Heiti 列中保留空白(space)}\label{tab:S:format}}
\sisetup{
    table-alignment-mode = format,
    table-number-alignment = center,
}
\begin{tabular}{@{}
    S[table-format = 2.2]
    S[table-format = 2.2, table-number-alignment = right]
    S[table-format = 2.2(1)]
    S[table-format = 2.2(1), separate-uncertainty]
    S[table-format = +2.2]
    S[table-format = 2.2e1]
    @{}
}
\hline{1.2pt}
    {\Heiti 值}
    & {\Heiti 值}
    & {\Heiti 值}
    & {\Heiti 值}
    & {\Heiti 值}
    & {\Heiti 值} \\
\hline{0.7pt}
```

```

2.3 & 2.3 & 2.3(5) & 2.3(5) & 2.3 & 2.3e8 \\
34.23 & 34.23 & 34.23(4) & 34.23(4) & 34.23 & 34.23 \\
56.78 & 56.78 & 56.78(3) & 56.78(3) & -56.78 & 56.78e3 \\
3,76 & 3,76 & 3,76(2) & 3.76(2) & +-3.76 & e6 \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 23: 在 s 列中保留空白 (space)

值	值	值	值	值	值
2.3	2.3	2.3(5)	2.3 ± 0.5	2.3	2.3 × 10 <sup>8</sup>
34.23	34.23	34.23(4)	34.23 ± 0.04	34.23	34.23
56.78	56.78	56.78(3)	56.78 ± 0.03	-56.78	56.78 × 10 <sup>3</sup>
3.76	3.76	3.76(2)	3.76 ± 0.02	±3.76	10 <sup>6</sup>

值得注意的是，没有在表格格式 (table format) 参数中指定的数字的任何部分都被设置为不存在 (absent)(数字的数量 [number of figures] 被设置为零)。设置 table-format (表格格式) 选项还可以将 table-alignment-mode (表格对齐模式) 重置为 format (格式化)。

通过将模型文本 (model text) 作为 table-format (表格格式) 键的一部分，可以保留 s 列前后的材料空白 (space for material)。然后用它来提供必要的间隙 (gap)，同时保持对齐 (maintaining alignment)(表 24)。

```

\begin{table}[H]
\caption{\ {\Heiti 数字}(numbers){\Heiti 前后的文本}(text)\label{tab:S:ends}}
\sisetup{table-format = {now }2.4{\textsuperscript{\emph{a}}}}
\begin{tabular}{@{}S@{}}
\hline{1.2pt}
{\Heiti 值} \\
\hline{0.7pt}
2.3456 \\
34.2345 \textsuperscript{\emph{a}} \\
56.7835 \\
now~ 90.473 \\
\hline{1.2pt}
\end{tabular}

```

`\end{table}`

表 24: 数字 (numbers) 前后的文本 (text)

值
2.3456
34.2345 <sup>a</sup>
56.7835
now 90.473

当在表格中打印指数 (exponents) 时, 可以选择对齐指数部分 (aligning the exponent parts) 或者让指数部分靠近尾数 (mantissa)。这是由 `table-align-exponent` (表格对齐指数) 选项控制的 (表 25)。同样, 与尾数分开打印的不确定性部分 (uncertainty parts) 可 `table-align-uncertainty` 以对齐 (aligned) 或接近 (closed up), 这是由 `table-align-uncertainty` (表格对列不确定性) 选项设置的 (表 26)。最后, 同样的方法也可用于带有 `table-align-comparator` (表格对齐比较符) 选项的比较符 (comparator)(表 27)。

```
\begin{table}[H]
\caption{\ \opt{table-align-exponent}\ {\Heiti 选项}\label{tab:align:exp}}
\sisetup{table-format = 1.3e2}
\begin{tabular}{@{}SS[table-align-exponent = false]@{}}
\hline{1.2pt}
{\Heiti 表头} & {\Heiti 表头} \\
\hline{0.7pt}
1.2e3 & 1.2e3 \\
1.234e56 & 1.234e56 \\
\hline{1.2pt}
\end{tabular}
\end{table}
```

表 25: `table-align-exponent` 选项

表头	表头
1.2 × 10 <sup>3</sup>	1.2 × 10 <sup>3</sup>
1.234 × 10 <sup>56</sup>	1.234 × 10 <sup>56</sup>

```

\begin{table}[H]
  \caption{\ \opt{table-align-uncertainty}\ {\Heiti 选项}\label{tab:align:uncert}}
  \sisetup{
    separate-uncertainty,
    table-format = 1.3(1),
  }
  \begin{tabular}{@{}SS[table-align-uncertainty = false]@{}}
    \hline[1.2pt]
      {\Heiti 表头} & {\Heiti 表头} \\
    \hline[0.7pt]
      1.2(1) & 1.2(3) \\
      1.234(5) & 1.234(5) \\
    \hline[1.2pt]
  \end{tabular}
\end{table}

```

表 26: table-align-uncertainty 选项

表头	表头
1.2 ± 0.1	1.2 ± 0.3
1.234 ± 0.005	1.234 ± 0.005

```

\begin{table}
  \caption{\ \opt{table-align-comparator}\ {\Heiti 选项}\label{tab:align:comp}}
  \sisetup{table-format = >2.2}
  \begin{tabular}{@{}SS[table-align-comparator = false]@{}}
    \hline[1.2pt]
      {\Heiti 表头} & {\Heiti 表头} \\
    \hline[0.7pt]
      > 1.2 & > 1.2 \\
      < 12.34 & < 12.34 \\
    \hline[1.2pt]
  \end{tabular}
\end{table}

```

表 27: table-align-comparator 选项

表头	表头
> 1.2	>1.2
<12.34	<12.34

注释标记 (note markers) 通常在数字内容 (numerical content) 之后的表格中给出。可能  
table-align-text-before 需要这些接近数字 (close up to the numbers)。这是否发生由 table-align-text-before  
table-align-text-after (env.) (在前面对齐表格文本) 和 table-align-text-before (在后面对齐表格文本) 选项控制  
(env.) (表 28)。

```
\begin{table}[H]
  \caption{\ {\Heiti 让注释接近文本}(closing notes up to text)\label{tab:S:notes}}
  \newrobustcmd\NoteMark[1]{%
    \textsuperscript{\emph{\#1}}}%
  }
  \sisetup{table-format = {\NoteMark{a}}2.4}
  \begin{tabular}{@{}
    S
    S[table-align-text-before = false]
  @{}}
    \hline[1.2pt]
      {\Heiti 值}          & {\Heiti 值} \\\
    \hline[0.7pt]
      2.3456 & 2.3456 \\\
      \NoteMark{a} 4.234 & \NoteMark{a} 4.234 \\\
      \NoteMark{b} .78 & \NoteMark{b} .78 \\\
      \NoteMark{d} 88 & \NoteMark{d} 88 \\\
    \hline[1.2pt]
  \end{tabular}
  \hspace{4em}%
  \sisetup{table-format = 2.4\NoteMark{a}}
  \begin{tabular}{@{}
    S
    S[table-align-text-after = false]
  @{}}
```

```

\hline{1.2pt}
{\Heiti 值} & {\Heiti 值} \\
\hline{0.7pt}
2.3456 & 2.3456 \\
34.234 \NoteMark{a} & 34.234 \NoteMark{a} \\
56.78 \NoteMark{b} & 56.78 \NoteMark{b} \\
90.4 \NoteMark{c} & 90.4 \NoteMark{c} \\
88 \NoteMark{d} & 88 \NoteMark{d} \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 28: 让注释接近文本 (closing notes up to text)

值	值	值	值
2.3456	2.3456	2.3456	2.3456
<sup>a</sup> 4.234	<sup>a</sup> 4.234	34.234 <sup>a</sup>	34.234 <sup>a</sup>
<sup>b</sup> 0.78	<sup>b</sup> 0.78	56.78 <sup>b</sup>	56.78 <sup>b</sup>
<sup>d</sup> 88	<sup>d</sup> 88	90.4 <sup>c</sup>	90.4 <sup>c</sup>
		88 <sup>d</sup>	88 <sup>d</sup>

表格单元格 (table cells) 的内容可以自动四舍五入或从零填充到 `table-format` (表格格式) 选项的小数部分 (decimal part) 所给的小数位数字 (number of decimal digits)。此 `table-auto-round (env.)` 模式使用 `table-auto-round` (表格自动舍入) 开关激活，如表 29 所示。

```

\begin{table}[H]
\centering
\caption{\ table-auto-round\ {\Heiti 选项}\label{tab:S:auto}}
\sisetup{table-format = 1.3}
\begin{tabular}{@{}SS[table-auto-round]@{}}
\hline{1.2pt}
{\Heiti 表头} & {\Heiti 表头} \\
\hline{0.7pt}
1.2 & 1.2 \\
1.2345 & 1.2345 \\
\hline{1.2pt}

```

```

\end{tabular}
\end{table}

```

表 29: table-auto-round 选项

表头	表头
1.2	1.200
1.2345	1.235

`parse-numbers` (*env.*) 当 `parse-numbers` (解析数字) 选项设置为 `false` 时, 表的对齐代码 (alignment code) 采用不同的方法。输出始终设置为数学模式 (math mode), 并且在第一个小数点标记 (decimal marker) 处进行对齐。这是通过在数学模式 (math mode) 下激活它来实现的。当为内容保留空白 (reserving space) 时, 只考虑尾数 (mantissa) 的整数 (integer) 和小数值 (decimal values)(表 30)。

```

\begin{table}[H]
\caption{\ {\Heiti 对齐}(aligning)\ {\Heiti 而不进行解析}(parsing)\label{tab:S:nonparsed}}
\sisetup{
  parse-numbers = false,
  table-format = 3.3
}
\centering
\begin{tabular}{@{}
S
S[table-number-alignment = center]
S[table-number-alignment = right]
S[table-number-alignment = left]
@{}}
\hline{1.2pt}
{\Heiti 一些值}
& {\Heiti 一些值}
& {\Heiti 一些值}
& {\Heiti 一些值} \\
\hline{0.7pt}
2.35 & 2.35 & 2.35 & 2.35 \\
34.234 & 34.234 & 34.234 & 34.234

```

```

56.783 & 56.783 & 56.783 & 56.783 \\
3,762 & 3,762 & 3,762 & 3.762 \\
\sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 30: 对齐 (aligning) 而不进行解析 (parsing)

一些值	一些值	一些值	一些值
2.35	2.35	2.35	2.35
34.234	34.234	34.234	34.234
56.783	56.783	56.783	56.783
3.762	3.762	3.762	3.762
$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$

在数据包含一系列值的情况下，在表中使用固定指数 (fixed exponent) 打印可以使显示 (presentation) 更清晰。在这些情况下，从表中删除 (dropping) 指数值是有用的。`drop-exponent (env.)` 一般的数值选项 (numerical options) `drop-exponent` (删除指数) 结合 `exponent-mode = fixed` 可以用来实现这一点 (表 31)。

```

\begin{table}[H]
\caption{\ \opt{drop-exponent}\ {\Heiti 选项}\label{tab:exp:omit}}
\begin{tabular}{@{}}
S[table-format = 1.1e1]
S[
drop-exponent = true,
exponent-mode = fixed,
fixed-exponent = 3,
table-format = 2.1,
]
@{}
\hline{1.2pt}
{\Heiti 表头} &
\multicolumn{1}{c@{}}{\Heiti 表头} / \num[print-unity-mantissa = false]{e3} \\
\hline{0.7pt}

```

```

1.2e3 & 1.2e3 \\
3e2 & 3e2 \\
1.0e4 & 1.0e4 \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 31: drop-exponent 选项

表头	表头 / $10^3$
$1.2 \times 10^3$	1.2
$3 \times 10^2$	0.3
$1.0 \times 10^4$	10

通常，数字列 (numerical column) 的宽度允许根据内容 (content) 的不同而变化。然而，有些情况下，一个严格固定的宽度 (strictly fixed width) 是可取的。对于这些情况，可以使用 `table-fixed-width` (表格固定宽度) 和 `table-column-width` (表格列宽) 选项。`table-fixed-width` (表格固定宽度) 激活固定宽度的列 (`fixed-width` `table-column-width` (env.) columns)，而 `table-column-width` (表格列宽) 定义目标宽度 (target width)(表 32)。将 `table-column-width` 设置为正值 (positive value) 将自动启用 `table-fixed-width`。

```

\begin{table}[H]
\caption{\ {\Heiti 固定宽度}(fixed-width){\Heiti 列}\label{tab:width:fixed}}
\begin{tabular}
{@{}
S
S[table-column-width = 2cm]
@{}}
\hline{1.2pt}
{Flexible} &
{Fixed}    \\
\hline{0.7pt}
1.23 & 1.23 \\
45.6 & 45.6 \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

`\end{table}`

表 32: 固定宽度 (fixed-width) 列

Flexible	Fixed
1.23	1.23
45.6	45.6

`table-column-width` (表格列宽) 选项也可用于实现特殊效果 (special effects)。一个例子是将一系列数字居中放置在宽标题 (wide heading) 下，数字本身右对齐 (表 33)。

```
\begin{table}[H]
  \centering
  \caption{\ {\Heiti 在标题下右对齐}\label{tab:width:special}}
  \settowidth{\mylength}{Long header}
  \sisetup{
    table-alignment-mode = none      ,
    table-column-width   = \mylength ,
    table-number-alignment = right
  }
  \begin{tabular}{@{}S@{}}
    \hline[1.2pt]
    {\Heiti 长表头} \\
    \hline[0.7pt]
      12.33 \\
      2      \\
      1234   \\
    \hline[1.2pt]
  \end{tabular}
\end{table}
```

表 33: 在标题下右对齐

长表头
12.33
2
1234

如图所示，可以使用大括号 (braces) 保护不属于数字的单元格内容。使用 `table-text-alignment (env.)` `table-text-alignment` (表格文本对齐) 选项指定的设置对齐完全不包含数字数据 (numerical data) 的单元格，该选项可识别 `center` (居中)、`left` (左)、`none` (无)、`right` (右) 等值 (表 34)。设置 `none` (无) 的目的是与宏包 `tabularray` 一起使用，该宏包它执行自己的文本值对齐 (alignment of textual values)。

```
\begin{table}[H]
  \caption{\ {\Heiti 在}~\texttt{S}~{\Heiti 列中对齐文本}\label{tab:S:text}}
  \sisetup{table-format = 4.4}
  \centering
  \begin{tabular}{@{}
    S
    S[table-text-alignment = left]
    S[table-text-alignment = right]
    @{}}
    \hline{1.2pt}
      {\Heiti 值}
      & {\Heiti 值}
      & {\Heiti 值} \\
    \hline{0.7pt}
      992.435 & 992.435 & 992.435 \\
      7734.2344 & 7734.2344 & 7734.2344 \\
      56.7834 & 56.7834 & 56.7834 \\
      3,7462 & 3,7462 & 3,7462 \\
    \hline{1.2pt}
  \end{tabular}
\end{table}
```

表 34: 在 s 列中对齐文本

值	值	值
992.435	992.435	992.435
7734.2344	7734.2344	7734.2344
56.7834	56.7834	56.7834
3.7462	3.7462	3.7462

表格对齐选项 (table alignment options) `table-number-alignment` (表格数字对齐) `table-alignment (env.)` 和 `table-text-alignment` (表格文本对齐) 设置的值可以与 `table-alignment` (表格对齐) 选项设置的值相同。这将会将所有三个对齐选项设置为相同的值 (`center`、`right` 或 `left`)。

### 4.13 局部设置选项

`siunitx` 允许用户通过使用局部 (locales) 设置在不同 (地理) 区域的排版约定 (typographic conventions) 之间进行切换。目前, 该宏包为 UK (英国)、US (美国)、DE (Germany, 德国)、PL (Poland, 波兰)、FR (French, 法国)、SI (Slovene, 斯洛文尼亚) 和 ZA (South Africa, 南非) 提供配置。 `locale (env.)` 选项用于切换到特定的区域 (particular locale)。

排版的结果	输入的命令
1.234 m	<code>\qty{1.234}{\metre}</code>
6,789 m	<code>\qty[locale = DE]{6.789}{\metre}</code>

### 4.14 仅在前言中使用的选项

`table-column-type (env.)` 可以使用 `table-column-type` (表格列类型) 选项调整用于创建表格列的字母 (letter)。标准设置为 `s`, 但可以使用一个或多个字母: 这些字母必须是单个令牌 (single tokens)。例如, 如果尚未加载 `numprint` 宏包, 则可以使用字母 `n`, 因为这会建议使用数字列 (numerical column)。

## 5 从版本 2 升级

该宏包在版本 2 和版本 3 之间进行了大量内部重写。大量键-值设置 (key-value settings) 都有新的、更具描述性 (descriptive) 的名称。在可能的情况下, 较旧的名称在内部映射到较新的名称: 如果是这种情况, 您将在日志中收到警告。

对于无法使用版本 3 成功处理的文档, 可以使用  $\LaTeX 2_{\epsilon}$  内核机制 (kernel mechanism) 加载最后一个版本 2 的发行版。这可以通过使用

```
\usepackage{siunitx}[=2021-04-09]
```

或

```
\usepackage{siunitx}[=v2]
```

来实现。这种方法将适用于仍然安装了版本 2 的旧系统, 这意味着您可以可靠地使用它在具有不同版本 `siunitx` 的系统之间工作。

---

<div style="display: flex; flex-direction: column; align-items: center;"><div><code>\SI</code></div><div><code>\SIl</code></div><div><code>\SIlist</code></div><div><code>\SIrange</code></div><div><code>\si</code></div></div>	<p>在版本 3 中, 文档命令 (document commands) 被修改为更具描述性。因此, 命令 <code>\SI</code>、<code>\SIl</code>、<code>\SIlist</code>、<code>\SIrange</code> 和 <code>\si</code> 仍然可用, 但不建议在新文档中使用。请改用新的 <code>\qty...</code> 命令: 它们更清晰, 在某些情况下速度稍快。</p>
--	--

---

对命令 (commands) 或选项的语义 (semantics) 进行了一些更改。最值得注意的是:

- 词头 (prefixes) 现在不能没有单位 (units)
- 词头 (prefixes) 只能作为数量 (quantity) 的一部分与数字 (numbers) 相互转换 (interconverted), 而不能作为独立的单位

如果要打印词头信息 (prefix information), 请参见第 9.16 节了解如何使用新方法。

字体控制系统 (font control system) 已为版本 3 完全重写。使用的方法与版本 2 完全不同。因此, 不能为所有结果 (outcomes) 提供仿真 (emulation): 如果您需要非标准字体设置 (non-standard font settings), 则需要调整源代码。请参阅第 4.2 节以获得有关此区域 (area) 中可用选项的更多详细信息。

与版本 2 相比, 版本 3 的输入方法 (input approach) 稍微更加结构化 (structured) 和受限制 (restricted)。除文档命令 (document commands) 更新后的名称 (updated names) 外, 这意味着:

- 现在必须使用专用的 `\numproduct` 和 `\qtyproduct` 命令给出数字的乘积 (products of numbers);
- 数字的商 (quotients of numbers) 仅支持作为文字 (literals);
- 复数值 (complex values) 需要使用专用命令 (dedicated command) `\complexnum` 给出。

`round-integer-to-decimal = false` 选项已被删除, 虽然没有直接替换, 但用户可能会发现 `round-pad = false` 提供了所需的结果 (outcome)。

已经采取了一种新的方法来提供用于单位的非拉丁符号 (non-Latin symbols): 这些符号现在可以在需要时直接处理, 例如在 `\micro` 词头 (prefix) 的定义中。

固定字符串 (fixed strings) 的翻译 (translations) 现在使用 `translations` 宏包进行。如果您已经使用 `translator` 在版本 2 中手动设置了翻译, 则需要手动加载。

用于数字表格列 (numerical tabular column) 的字母现在可以由用户选择: 字母 `s` 保留为标准接口 (standard interface)。单位列 (unit column)(`s`) 已从此版本中删除。例如, 可以使用 `colcell` 宏包对其进行模拟 (emulated)

```
\usepackage{colcell}
\newcolumnntype{s}{>\collectcell\unit}c<\endcollectcell}}
```

或

```
\usepackage{colcell}
\newcolumnntype{s}{>\collectcell\si}c<\endcollectcell}}
```

如果使用 `table-column-width` (表格列宽) 来设置固定宽度的列 (fixed-width columns), 那么现在还需要 `table-fixed-width` (表格固定宽度) 将此选项设置为活动的 (active)。

已删除对加载局部配置文件 (local configuration file) `siunitx.cfg` 的直接支持。然而, 可以使用第 9.17 节中描述的方法, 通过在文档源 (document sources) 中更清楚地显示添加内容来达到同样的效果。

命令 `\SendSettingsToPgf` 已被弃用, 应该通过设置与 `\sisetup` 并行的适当 `\pgfkeys` 来替换该命令。

## 6 BIPM 进行的单位变更

除了第 5 节中描述的 `siunitx` 宏包的变化外，第 9 版 `si` 手册 (`si Brochure`) 中 `bipm`<sup>5</sup> 定义的单位也有变化，这些变化反映在这里。有两个主要的变化领域。

第一个是关于可与 `si` 单位一起使用的单位。在第 8 版 `si` 手册 (`si Brochure`) 中，下列单位被列为可用于专业领域 (`specialist fields`)

- `ångström` (`\angstrom`), 即长度单位埃米。
- `bar` (`\bar`), 即压强单位巴。
- `barn` (`\barn`), 即面积单位靶恩。
- `knot` (`\knot`), 即速率单位节。
- `millimetre of mercury` (`\mmHg`), 即压强单位毫米汞柱。
- `nautical mile` (`\nauticalmile`), 即长度单位海里。

这些在第 9 版中不再列出，因此 `siunitx` 不推荐将其作为预定义单位 (`pre-defined units`)。这些单位在首次使用时将发出警告，用户应在其源代码的开头添加自己的定义以避免这种情况。

其次，对基本单位 (`base units`) 的新定义 (`new definition`) 意味着实验确定的单位表 (`table of units`) 已从 `si` 手册 (`si Brochure`) 中删除。这包括 `siunitx` 在早期版本中定义的以下单位：

- `\bohr`
- `\clight`
- `\electronmass`
- `\elementarycharge`
- `\hartree`
- `\planckbar`

---

<sup>5</sup>`bipm`: 法语 *Bureau international des poids et mesures* 的缩写，国际计量局

这些在 `siunitx` 中也不被推荐，用户应该提供自己的定义。

除了这两个主要模块 (major blocks) 外，单位 `\atomicmassunit` 也有类似的弃用状态 (deprecated status)：在第 8 版 si 手册 (si Brochure) 中，它与实验确定的单位 (experimentally-determined units) 相同，但与道尔顿 (dalton) 相当，这一单位仍被接受。

## 7 本地化

translations 宏包为单词 (words) 和短语 (phrases) 的本地化 (localisation) 提供了一个结构化框架 (structured framework)。特别是，它提供了 `\GetTranslation` 宏，该宏将根据当前的 babel 或 polyglossia 语言设置 (language setting) 提供适当的翻译 (translations)。

如果 translations 可用，siunitx 将加载它并更改 `list-final-separator` (列表最终分隔符) 和 `range-phrase` (范围短语) 选项的标准设置 (standard settings)，以读取：

```
\sisetup{
  list-final-separator = { \GetTranslation{and} },
  list-pair-separator  = { \GetTranslation{and} },
  range-phrase         = { \GetTranslation{to (numerical range)} },}
```

如果 translations 宏包知道当前语言 (current language)，那么结果将是本地化文本 (localised text)。本手册的前言加载了英语 (English)、法语 (French)、德语 (German)、波兰语 (Polish)、西班牙语 (Spanish)、加泰罗尼亚语 (Catalan)、葡萄牙语 (Portuguese) 和巴西语 (Brazilian) 作为选项，还加载了 babel 宏包：

排版的结果	输入的命令
1, 2 and 3	<code>\numlist{1;2;3}</code> % 默认为英语 (English)
1 to 10	<code>\numrange{1}{10}</code> % 默认为英语 (English)
1, 2 et 3	<code>\selectlanguage{french}\numlist{1;2;3}</code>
1 à 10	<code>\selectlanguage{french}\numrange{1}{10}</code>
1, 2 und 3	<code>\selectlanguage{german}\numlist{1;2;3}</code>
1 bis 10	<code>\selectlanguage{german}\numrange{1}{10}</code>
1, 2 i 3	<code>\selectlanguage{polish}\numlist{1;2;3}</code>
1 do 10	<code>\selectlanguage{polish}\numrange{1}{10}</code>
1, 2 y 3	<code>\selectlanguage{spanish}\numlist{1;2;3}</code>
1 a 10	<code>\selectlanguage{spanish}\numrange{1}{10}</code>
1, 2 i 3	<code>\selectlanguage{catalan}\numlist{1;2;3}</code>
1 a 10	<code>\selectlanguage{catalan}\numrange{1}{10}</code>
1, 2 e 3	<code>\selectlanguage{portuguese}\numlist{1;2;3}</code>
1 a 10	<code>\selectlanguage{portuguese}\numrange{1}{10}</code>
1, 2 e 3	<code>\selectlanguage{brazilian}\numlist{1;2;3}</code>
1 a 10	<code>\selectlanguage{brazilian}\numrange{1}{10}</code>

## 8 与其他宏包的兼容性

一般来说, `siunitx` 应该可以与其他宏包一起使用而不会受到干扰 (interference)。

在 `siunitx` 之前加载 `physics` 宏包时, 没有定义 `\qty` 命令。用户可以使用版本 2 命令 `\SI`, 它可以用来替代 `\qty`。或者, 如果您更喜欢 `siunitx` 定义, 则可以使用

```
\AtBeginDocument{\RenewCommandCopy\qty\SI}
```

并使用较长的名称 (longer name) `\quantity` 来访问 `physics` 宏包的功能。

## 9 使用 siunitx 时的注意事项

### 9.1 有问题的字体编码

siunitx 中的标准设置 (standard settings) 假定以“合理 (sensible)”输入和字体编码值 (font encoding values) 为准。在所有情况下, 假设输入编码 (input encoding) 为 utf-8。对于 pdf $\LaTeX$ , 字体编码应为 T1, 而对于 Xe $\LaTeX$  和 Lua $\LaTeX$ , 则应为 TU(Unicode 字体编码)。

一些宏包, 例如 newtxtext 或 stix2, 要么强制 T1, 要么不能正确预测 Xe $\LaTeX$  和 Lua $\LaTeX$  的 TU。在这些情况下, siunitx 使用的符号 (symbols) 可能不正确。如果无法更正字体编码 (font encoding), 则需要使用解释此非标准设置 (non-standard setup) 的符号定义 (symbol definitions) 重新声明相关单位 (relevant units)。

### 9.2 调整 \litre 和 \liter

如前所述, 单位宏 (unit macros) \litre 和 \liter 都可用于升 (litres)。使用标准设置 (standard settings), \liter 定义为

```
\DeclareSIUnit\liter{\litre}
```

这意味着 \litre 是“规范 (canonical)”单位。这与 \metre 和 \meter 之间的关系相同。

然而, 与米 (metres) 相反, 用户更可能希望调整升的外观: “l”和“L”都常用。建议的调整方法是重新声明 \litre 宏, 因为 \liter 将自动跟随 (follow automatically)。

```
\DeclareSIUnit\litre{l}
```

### 9.3 确保文本或数学输出

应使用宏 \ensuremath 和 \text 确保特定条目 (item) 始终以所需模式 (desired mode) 打印。一些数学输出 (mathematical output) 在 \mathrm (siunitx 用于打印单位的字体设置) 中不正常。解决这一问题的最简单方法是使用构造 (construction) \text{\ensuremath{...}}, 它将以标准数学字体 (standard mathematics font) 打印材料, 而不会影响其余输出。在某些情况下, 简单地强制使用 \mathnormal 就足够了, 但这对于非拉丁字符 (non-Latin characters) 来说不太可靠。

## 9.4 在 `\text` 中包含文字连字符

在大多数情况下，`\text` 内的 “-” 字符将表示减号 (minus)。因此，宏包将用 `\textminus` 替换它。然而，如果您需要连字符 (hyphen)，或者在  $\text{\TeX}$  语法中使用 “-”，这可能会有问题。可以通过使用第二组大括号 (braces) 来防止这种情况：

```
\DeclareSIUnit\electronvolt{\text{\e\kern -0.1em V}}
```

或者定义一个受保护的命令 (protected command)，该命令在排版时会产生连字符 (hyphen)：

```
\usepackage{etoolbox}
\newrobustcmd*\hyphenminus{-}
\DeclareSIUnit\electronvolt{\text{\e\kern \hyphenminus0.1em V}}
```

## 9.5 展开表格中的内容

处理表格 (tables) 时，`siunitx` 将展开宏中存储的任何内容，除非它是长的 (long) 或受保护的 (protected)。还检测到  $\text{\LaTeX 2}_{\epsilon}$  鲁棒命令 (robust commands)，且未进行展开 (表 35)。否则将展开的值可以通过将它们包在一组大括号中来保护。由于  $\text{\TeX}$  本身会在 `siunitx` 对表格单元格中的第一个令牌 (token) 进行操作之前展开它，因此建议使用  $\epsilon\text{-}\text{\TeX}$  保护机制 (protected mechanism) 来防止表格单元格中宏的扩展。如表所示， $\text{\TeX}$  对  $\text{\LaTeX 2}_{\epsilon}$  鲁棒命令的展开可能导致意外结果 (unexpected results)。

```
\begin{table}[H]
  \caption{\ {\Heiti 值作为}~\texttt{S}~\ {\Heiti 列中的宏}\label{tab:xmpl:macro}}
  \newcommand*\myvaluea{1234}
  \newcommand\myvalueb{1234}
  \DeclareRobustCommand*\myvaluec{1234}
  \protected\def\myvalued{1234}
  \begin{tabular}{@{}S@{}}
    \hline{1.2pt}
      {\Heiti 一些值} \\\
    \hline{0.7pt}
      \myvaluea 8.8 \myvaluea \\\ % 两者均已展开
      \myvalueb 8.8 \myvalueb \\\ % 首先由 TeX 展开为数字
      \myvaluec 8.8 \myvaluec \\\ % 首先由 TeX 展开，但不是数字！
      \myvalued 8.8 \myvalued \\\ % 均未展开
  \end{tabular}
\end{table}
```

```

{\myvaluea\ 8.8 \myvaluea} \\\ % 均未展开
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 35: 值作为 s 列中的宏

一些值
12 348.812 34
12 348.812 34
12348.81234
12348.81234
1234 8.8 1234

可以在表格 (tables) 中使用计算值 (calculated values)。要使其工作，必须在尝试解析数字 (parse the number) 之前进行计算 (解析器无法“知道 [know]”表达式中的所有可能值)。这是使用 xfp 宏包最方便地处理的，xfp 宏包作为 siunitx 所需支持的一部分分发。一般方法 (general approach) 如表 36 所示。

```

\begin{table}[H]
\caption{\ {\Heiti 已计算的值}\label{tab:xmpl:calc}}
\newcommand{\valuea}{66.7012}
\newcommand{\valueb}{66.0212}
\newcommand{\valuec}{64.9026}
\begin{tabular}{@{}
S[table-format = 2.4]
S[table-format = 3.4]
@{}
}
\hline{1.2pt}
{\Heiti 值} & {\Heiti 双倍的} \\\
\hline{0.7pt}
\valuea & \fpeval{\valuea * 2} \\\
\valueb & \fpeval{\valueb * 2} \\\
\valuec & \fpeval{\valuec * 2} \\\

```

```

\hline{1.2pt}
\end{tabular}
\end{table}

```

表 36: 已计算的值

值	双倍的
66.7012	133.4024
66.0212	132.0424
64.9026	129.8052

一种更复杂的方法 (sophisticated approach) 是从数据库 (database) 生成行本身 (rows themselves): 第 9.6 节对此进行了说明。

## 9.6 将 siunitx 与 datatool 一起使用

如表 36 所示, siunitx 可用于对使用 datatool 存储的数据 (data) 进行排版。为了快速显示表格 (tables) 的内容, datatool 提供了 \DTLshowtable 宏。这仅在关闭数字解析 (number parsing) 时才能使用 S 列 (表 37)。

```

\DTLnewdb{moredata}
\DTLnewrow{moredata}\DTLnewdbentry{moredata}{value}{ 6.7012}
\DTLnewrow{moredata}\DTLnewdbentry{moredata}{value}{66.0212}
\DTLnewrow{moredata}\DTLnewdbentry{moredata}{value}{64.902 }
\begin{table}
\caption{\ {\Heiti 显示}~\textsf{datatool}~\ {\Heiti 表}\label{tab:xmpl:datatool}}
\sisetup{parse-numbers= false, table-format = 2.4}
\renewcommand*{\dtlrealalign}{S}
\DTLdisplaydb{moredata}
\end{table}

```

datatool 宏包也可以用于使用计算 (calculations) 创建动态表 (on-the-fly tables)。例如, 表 36 中的演示 (demonstration) 可以在不需要写出每一行的情况下实现, 如表 38 所示。在这种情况下, 额外的一列 (extra column) 用于包含计算 (calculations)。

```

\begin{table}[H]
\caption{\ {\Heiti 使用}~\pkg{datatool}~\ {\Heiti 计算的值}\label{tab:xmpl:datatool-calc}}

```

表 37: 显示 datatool 表

**value**  
6.7012  
66.0212  
64.902

```
\DTLnewdb{data}
\DTLnewrow{data}\DTLnewdbentry{data}{value}{66.7012}
\DTLnewrow{data}\DTLnewdbentry{data}{value}{66.0212}
\DTLnewrow{data}\DTLnewdbentry{data}{value}{64.9026}
\begin{tabular}{
  @{}
  S[table-format = 2.4]
  S[table-format = 3.4]
  @{}l
  @{}
}
\hline{1.2pt}
  {\Heiti 值} & {\Heiti 双倍的} &
\DTLforeach{data}{\myvalue=value}{%
  \DTLiffirstrow {\ \hline{0.7pt}}{\ \}%
  \myvalue & % First column
  \fpeval{\myvalue * 2} % second column
  & }\ \
\hline{1.2pt}
\end{tabular}
\end{table}
```

表 38: 使用 datatool 计算的值

值	双倍的
66.7012	133.4024
66.0212	132.0424
64.9026	129.8052

## 9.7 在节标题和书签中使用单位

siunitx 代码被设计用于正确处理标题 (headings) 中的函数 (functions)。它们将在标题和目录 (table of contents) 中正确打印。如这里所示, 标准行为 (standard behaviour) 是忽略字体更改 (font changes)。当 hyperref 宏包被加载时, 这些函数会自动 “优雅地降级 (degrade gracefully)”, 在 pdf 书签 (bookmarks) 中生成有用的信息。如果希望对书签文本 (bookmark text) 进行更多控制, 请使用 hyperref 中的 `\texorpdfstring` 函数, 例如:

```
\section{Some text
  \texorpdfstring
    {\unit{\joule\per\mole\per\kelvin}}
    {J mol-1 K-1}%
}
```

您可能会发现加载带有 `unicode` 选项的 hyperref 宏包很有用, 因为这将允许  $\Omega$  显示在书签 (bookmarks) 中。如果没有 `unicode` 选项, hyperref 使用的编码 (encoding) 不支持此符号 (symbol)。

## 9.8 在标题下视觉居中 (visually centred) 的左对齐列

当你有一列不相关的数字 (non-related numbers) 时, 通常的建议是将这些数字左对齐 (left-aligned), 然后将结果列 (resulting column) 居中放置在标题下。对于 dcolumn 宏包, 这将使用类似于 `D{x}{5.0}` 的方法来完成。这有点滥用数字的性质, 但也可以使用 siunitx 来完成 (表 39)。

```
\begin{table}[H]
  \caption{\ \Heiti 格式化不相关的数字\label{tbl:xmpl:unrel}}
  \centering
  \begin{tabular}
    {
      @{}
      S[
        table-format = 5.0,
        parse-numbers = false,
        input-decimal-markers = x
      ]
      @{}
    }
  \end{tabular}
\end{table}
```

```

\hline{1.2pt}
\multicolumn{1}{@{}c@{}}{\Heiti 表头} \\
\hline{0.7pt}
120   \\
12.3  \\
12340 \\
12.02 \\
123   \\
1     \\
\hline{1.2pt}
\end{tabular}
\end{table}

```

表 39: 格式化不相关的数字

表头
120
12.3
12340
12.02
123
1

## 9.9 回归表

在一些学科领域 (subject areas), 通常会显示回归值 (regression values) 或类似值, 其特征是在主值 (main value) 下方的括号中显示不确定性值 (uncertainty value)。由于这些单元格是独立的, 因此不能在一个值中使用 `siunitx` 输入它们。根据是否还需要解析 (parsed) 这些值, 有几种方法可以使用宏包对它们进行格式化。

在不需要解析 (parsing) 的地方, 可以使用最直接的方法: 提供一种模型格式 (model format), 允许在每个结尾处 (each end) 留出一个额外的“数字 (digit)”空间 (space), 然后允许使用括号 (parenthesis)。如果在数字 (number) 上加上符号 (sign), 则可能无需在前导括号 (leading bracket) 中添加“数字 (digit)”。如果还需要解析, 则不能采用这种方法。相反, 需要对解析进行调整, 使 ( 和 ) 不被视为数字 (number) 的一部分, 并且将

`table-align-text-before` (在前面对齐表格文本) 设置为 `false`, 以便将它们放在数字部分 (numerical part) 的旁边。这些方法如表 40 所示。

```
\begin{table}[H]
  \caption{\ {\Heiti 回归表}(regression tables)\label{tab:regression}
}
  \begin{tabular}
    {
      @{}
      S[table-format = 2.4, parse-numbers = false]
      S[table-format = +1.4, parse-numbers = false]
      S[
        input-open-uncertainty = ,
        input-close-uncertainty = ,
        minimum-decimal-digits = 3, % (
        table-format           = +1.3),
        table-align-text-before = false
      ]
      @{}
    }
  \hline{1.2pt}
  {\Heiti 表头} & {\Heiti 表头} & {\Heiti 表头} \\
  \hline{0.7pt}
  1.234   & -1.234 & -1.23   \\
  (0.053) & (0.053) & (0.053) \\
  \hline{1.2pt}
\end{tabular}
\end{table}
```

表 40: 回归表 (regression tables)

表头	表头	表头
1.234	−1.234	−1.230
(0.053)	(0.053)	(0.053)

## 9.10 最大化性能

在  $\text{\TeX}$  编程方面, 数字 (number) 和单位解析器 (unit parsers) 都需要大量的工作。对于不需要这种处理 (processing) 的输入, 可以通过关闭两个系统来获得 `siunitx` 的最大性能 (maximum performance):

排版的结果	输入的命令
7.3 Hz	<code>\qty{7.3}{\Hz}</code>
7.3 Hz	<code>\qty[parse-units=false]{7.3}{\Hz}</code>
7.3 Hz	<code>\qty[parse-numbers=false,parse-units=false]{7.3}{\Hz}</code>

对于表格 (tables), 在表格之前可以给出的任何设置都只能解析一次, 而在 `s` 的可选参数 (optional argument) 中, 则在每个单元格 (cell) 中读取这些设置。因此, 对于通用设置 (common settings), 您应该首选:

```
\begin{table}[H]
  \sisetup{...}
  \begin{tabular}{S}
    ...
```

## 9.11 对 `\kWh` 单位的特殊考虑

标准设置 (standard settings) 提供了 `\kWh` 单位设置 (unit set), 在 “kW” 和 “h” 单位之间没有间隔 (spacing), 从而得到 “kWh”。但是, 这只适用于单位是自己给出的: 组合 (combinations) 将遵循正常的规则 (normal rules)

排版的结果	输入的命令
kWh	<code>\unit{\kWh}</code>
$\text{kW h m}^{-1}$	<code>\unit{\kWh\per\metre}</code>

这是因为定义了单位 `\kWh`, 所以它仍然可以通过改变 `\kilo`、`\watt` 和 `\hour` 来改变, 因此词头 (prefix) 仍然可以变成数字 (number)。然而, 一些用户可能更喜欢使用从不添加空格 (space) 的非灵活宏 (non-flexible macro)。这可以通过使用 `\DeclareSIUnit` 重新定义 `\kWh`、添加替代定义 (alternative definition)

```
\DeclareSIUnit\kWh{kWh}
\DeclareSIUnit\KWH{kWh}
```

或使用文字单位输入 (literal unit input) 来实现。

排版的结果	输入的命令
$\text{kWh m}^{-1}$	<code>\unit{\KWH\per\metre}</code>
$\text{kWh m}^{-1}$	<code>\unit{kWh.m^{-1}}</code>

另一点需要注意的是，`\per` 适用于下一个单位 (next unit)，而不是整个单位组合 (unit combination)。因此，在

排版的结果	输入的命令
$\text{cd kW}^{-1} \text{ h}$	<code>\unit{\candela\per\kW}</code>

中 `\per` 适用于瓦特 (watts)，但不适用于小时 (hours)。在这种情况下，需要将单位完整地写出来，否则应使用 `sticky-per` (粘着每) 选项。

排版的结果	输入的命令
$\text{cd kW}^{-1} \text{ h}^{-1}$	<code>\unit{\candela\per\kilo\watt\per\hour}</code>
$\text{cd kW}^{-1} \text{ h}^{-1}$	<code>\unit[sticky-per]{\candela\per\kW}</code>

## 9.12 创建包含数字和单位的列

通常，表格中的数字 (numbers) 应与列标题 (column heading) 中的单位一起给出。然而，在某些情况下，一系列数据 (series of data) 最好以表格形式呈现，但单位不同。有两种方法可以做到这一点 (表 41)。第一种方法是将单位放在表格的第一列，如果表中有几个相关的条目 (related items)，这是有意义的。第二种方法是生成两列，一列用于数字，另一列用于单位，然后对它们进行格式化，以获得单个列的视觉效果 (visual effect)。当表格中只显示一组数字时，后一种效果最为合适。此方法需要收集单元格内容 (cell content)，使用 `colcell` 宏包最容易做到。

```
\begin{table}[H]
\caption{\ \Heiti 表格中的数字具有不同的单位\label{tab:xmpl:mixed}}
\hspace{0em}%
\begin{tabular}
{
```

```

@{}
>{\$}l<{\$}
S[table-format = 3.3(1)]
S[table-format = 3.3(1)]
@{}
}
\hline{1.2pt}
& {\Heiti 第一列} & {\Heiti 第二列} \\\
\hline{0.7pt}
a / \unit{\pm} & 123.4(2) & 567.8(4) \\\
\beta / \unit{\degree} & 90.34(4) & 104.45(5) \\\
\mu / \unit{\per\mm} & 0.532 & 0.894 \\\
\hline{1.2pt}
\end{tabular}
\hspace{3em}%
\begin{tabular}
{
@{}
S[table-format=1.3]@{\,,}
>\collectcell\unit}l<\endcollectcell}
@{}
}
\hline{1.2pt}
\multicolumn{2}{@{}c}{\Heiti 表头} \\\
\hline{0.7pt}
1.234 & \metre \\\
0.835 & \candela \\\
4.23 & \joule\per\mole \\\
\hline{1.2pt}
\end{tabular}
\hspace{\fill}%
\end{table}

```

表 41: 表格中的数字具有不同的单位

	第一列	第二列	表头
$a/\text{pm}$	123.4(2)	567.8(4)	1.234 m
$\beta/^\circ$	90.34(4)	104.45(5)	0.835 cd
$\mu/\text{mm}^{-1}$	0.532	0.894	4.23 J mol <sup>-1</sup>

9.13 带有标题行的表格

表格的常见格式 (common format) 是使用背景色 (background color) 和粗体文本 (bold text) 使标题行 (heading row) 在视觉上不同。如果数字 (numbers) 出现在 S 列中的标题行中, 那么正确的显示可能是一个挑战。最好的方法是使 `\bfseries` 宏 “健壮 (robust)” (如第 9.5 节所示), 然后使用该宏使标题单元格 (heading cells) 加粗。表 42 说明了这种方法, 以及使用 `\rowcolor` 提供背景色。

```
\begin{table}[H]
  \caption{\ \Heiti 表格中的标题行\label{tab:xmpl:headers}}
  \robustify\bfseries
  \centering
  \begin{tabular}
    {@{}S[text-series-to-math, table-format = 3.3]@{}}
    \rowcolor[gray]{0.9}
    \bfseries 123.456 \\\
    23.45 \\\
    123.4 \\\
    3.456 \\\
  \end{tabular}
\end{table}
```

表 42: 表格中的标题行

<b>123.456</b>
23.45
123.4
3.456

## 9.14 将区域设置与 babel 语言关联

在更改语言 (changing language) 时, 可以指示 (instruct) babel 宏包切换到特定的 siunitx 语言环境 (locale)。这可以使用 babel \extras{*language*} 系统完成。例如, 要将 DE 与 german babel 相关联, 适当的代码应该是:

```
\addto\extrasgerman{\sisetup{locale = DE}}
```

## 9.15 符号“数字 (digits)”

在某些情况下, 您可能希望使用不属于通常设置 (usual set) 0123456789 的“数字 (digits)”。这可以通过设置 input-digits (输入数字) 选项来实现, 但是要记住, 这会影响 (防止) 例如舍入 (rounding)。

排版的结果	输入的命令
$4\pi \times 10^{-7}$	<code>\sisetup{input-digits = 0123456789\pi}\num{4\pi e-7}</code>

每个额外的条目 (extra entry) 都应该是一个单独的令牌 (single token), 并且应该有一个在数学模式 (math mode) 和文本模式 (text mode) 下都安全的定义, 或者只能在输出模式 (output mode) 已知时使用。可能需要使用 etoolbox 使某些令牌变得健壮, 以使其工作, 例如:

排版的结果	输入的命令
$0.4066\dots\text{m}^2$	<code>\robustify\dots\sisetup{input-digits=0123456789\dots}</code> <code>\qty{0,4066\dots}{\metre\squared}</code>

## 9.16 演示词头

由于 siunitx 包含有关单位词头 (unit prefixes) 数值的数据, 您可能希望以自动方式 (automated way) 打印这些数据。词头不能单独给出, 但是可以创建一个“什么都不做 (do nothing)”单位。

```
\DeclareSIUnit\noop{\relax}
```

然后它可以用来仅显示词头符号 (prefix symbol)。

排版的结果	输入的命令
Y	<code>\unit{\yotta\noop}</code>

要仅显示词头的数值 (value of a prefix), 您需要使用 `\qty` 和适当的设置 (appropriate settings)。

排版的结果	输入的命令
$10^{24}$	<code>\qty[prefix-mode=extract-exponent, print-unity-mantissa=false]{1}{\yotta\noop}</code>

这可以方便地封装在文档命令 (document command) 中, 例如:

```
\NewDocumentCommand\prefixvalue{m}{%
  \qty[prefix-mode=extract-exponent,print-unity-mantissa=false]{1}{#1\noop}
}
```

## 9.17 创建一组预定义的单位

有许多单位位于 (当前) si 手册 (si Brochure) 中定义的单位之外, 这些单位对许多人都有用。最明显的是那些已经在手册的前几版中详细描述, 如第 6 节所述, 但还有很多其他的。

有一组预定义的 (pre-defined) 有用单位可用, 而无需将完整的定义复制到每个源文件 (source file) 中, 这通常很方便。同时, 重要的是, 这些来源 (sources) 确实表明它们使用的是 siunitx 核心部分未定义的单位。实现这一点的最直接的方法是创建一个单独的文件, 例如 `siunitx-local-units.tex`, 并将其放置在本地  $\TeX$  树中 (在 Linux 中通常为 `~/texmf/tex/latex/`, 在 macOS 中通常为 `~/Library/texmf/tex/latex/`, 在 Windows 中通常为 `C:\Users\<name>\texmf\tex\latex`)。然后在前言中可以使用下面的语句加载:

```
% Load useful 'local' units
\input{siunitx-local-units}
```

## 10 使用 (SI) 单位

一致的符合逻辑的单位是科学工作的必要条件，在任何地方都具有适用性。历史上，许多系统被用于物理单位 (physical units)。si 单位是 1960 年由法国计量学会 (*Conférence Générale des Poids et Mesures*, cgpm) 引入的。si 单位是一个基于七个基本单位 (base units) 的相干系统 (coherent system)，所有其他单位都可以从中导出。

同时，带单位的物理量 (physical quantities) 是数学实体 (mathematical entities)，因此，它们的排版很重要。在数学中，字体的变化 (如使用粗体、斜体、无衬线字体等) 传达信息。这意味着，规则 (rules) 不仅适用于 si 系统下使用的单位类型 (type of units)，也适用于它们在打印中的显示方式。国家标准与技术研究院 (*National Institute of Standards and Technology*, nist) [2] 提供了关于最佳做法的建议。

为了符合公认的国际标准 (international standard)，详细介绍了全部规则。这里不适合重复全部这些内容，相反，这里提供了一个有用的关键点摘要。详细信息可从国际计量局 (*Bureau International des Poids et Mesures*, bipm) [1] 获得。

siunitx 宏包尽可能地考虑了这里提供的信息。因此，宏包默认遵循为排版数字 (numbers) 和单位 (units) 所做的建议。间距 (spacing) 等等的处理方式使规则的实现 (相对) 容易。

### 10.1 单位

表 1 中<sup>6</sup>列出了七个基本 si 单位。基本单位的选择使得所有物理量都可以使用这些单位的适当组合 (appropriative combination) 来表示，而不需要其他单位，也没有冗余 (redundancy)。

si 系统中的所有其他单位被认为是从七个基本单位“派生 (derived)”出来的。在最基本的情况下，所有其他 si 单位都可以表示为基本单位的组合 (combinations of the base units)。但是，许多单位 (见表 2) 有一个特殊的名称 (name) 和符号 (symbol)。这些单位中的大多数是一个或多个基本单位的简单组合 (视情况提高到幂)。

提供了一系列用于十进制倍数 (decimal multiples) 和子倍数 (sub-multiples) 的 si 词头 (prefixes)，可用作除千克 (kilogram) 以外的任何 si 单位 (基本单位或派生单位) 的修饰符 (modifiers)。表 4 列出了词头。词头和单位之间不应该使用空格 (space)，只能使用单个词头 (single prefix)。即使是摄氏度 (degree Celsius) 也可以加上词头，例如 1 m°C。

---

<sup>6</sup>有些基本单位需要先定义其他单位，因此有一个必要的定义顺序 (order of definition)。

需要注意的是，千克 (kilogram) 是唯一一个在其名称 (name) 和符号 (symbol) 中带有词头 (prefix) 的 si 单位。只能使用单个词头 (single prefix)，因此在千克词头 (kilogram prefix) 的情况下，名称 (names) 与单位名称 (unit name) “克 (gram)” 一起使用，词头符号 (prefix symbols) 与单位符号 (unit symbol) g 一起使用。例如， $1 \times 10^{-6} \text{ kg} = 1 \times 10^{-3} \text{ g} = 1 \text{ mg}$ 。

si 单位的应用旨在提供一组单位，确保所有领域的一致性 (consistency) 和清晰度 (clarity)。然而，其他单位在许多领域都很常见，并不是一无是处。默认情况下，siunitx 提供的单位不包括任何这些单位，仅定义作为 si 集合 (set) 的一部分或可接受与 si 单位一起使用的单位。但是，其他几组单位可以作为可选模块 (optional modules) 加载。二进制词头 (binary prefixes) 和单位 (表 6) 是最明显的例子。这些不是 si 规范 (specifications) 的一部分，但词头名称 (prefix names) 源自表 4 中的词头名称。

在可能的情况下，通常应避免使用其他单位。基本上，si 单位应该是首选的，因为它具有清晰的定义 (clear definition) 和自一致性 (self-consistency) 的优点。然而，可能总是有专门 (specialist) 或非标准单位 (non-standard units) 的位置。从基本物理常数 (basic physical constants) 导出的单位尤其如此。

在许多领域，非标准单位的使用非常普遍，以至于很难或不可能做其他事情。例如，大多数合成化学家 (synthetic chemists) 测量真空设备 (vacuum apparatus) 内的压力，单位为 mmHg，部分原因是最常用的压力计仍然使用金属汞柱。出于这些原因，siunitx 允许定义此类单位。

## 10.2 数学意义

如前所述，数量组合 (quantity combination) 是一个单一的数学实体 (single mathematical entity)。这对于如何打印数字 (number) 和单位 (unit) 都有影响。首先，这两个部分不能分开：一个数量 (quantity) 是数 (number) 和单位 (unit) 的乘积 (product)。除了表示平面角度 (plane angles) 的符号 ( $^{\circ}$ 、 $'$  和  $''$ )，bipm 指定应该使用空格 (space) 或半高 (中心) 点 [1]。

排版的结果	输入的命令
A space for 10 %	<code>A space for \qty{10}{\percent}</code>
and also for 100 °C	<code>and also for \qty{100}{\degreeCelsius}</code>
but not for 1.23°	<code>but not for \ang{1.23}</code>

单位的数学意义 (mathematical meaning) 也意味着形状 (shape)、权重 (weight) 和族

(family) 很重要。单位应该用一种直立 (upright)、中等粗细的 (medium weight) 字体排版。斜体 (italic)、粗体 (bold) 和无衬线 (sans serif) 在数学上都用来表达其他含义。在全无衬线字体 (all sanserif) 文档中，使用无衬线字体 (sans serif) 表示单位是合理的。siunitx 宏包默认再次遵循此约定：忽略任何本地设置 (local settings)，并使用当前的直立数学字体 (upright math font)。然而，在某些情况下，这可能不是最理想的行为。一个典型的例子是在一个全粗体的节标题 (all-bold section heading) 中，由于周围的文字是粗体的 (bold)，一些人认为任何单位都应该遵循这一点。

排版的结果	输入的命令
Units should <b>not be bold</b> : 54 F	Units should \textbf{not be bold: \qty{54}{\farad}}
<b>But perhaps in a running block, it might look better:</b> 54 F	\textbf{But perhaps in a running block, \textbf{it might look better: } \qty[text-series-to-math]{54}{\farad}}

由其他单位乘法 (multiplication) 形成的单位的符号 (symbols) 用半高 (即中心) 点或 (细) 空格表示。

排版的结果	输入的命令
m s = metre second	\$\unit{\metre\second} = \text{metre second}\$
ms = millisecond	\$\unit{\milli\second} = \text{millisecond}\$
m · s = metre second	\sisetup{inter-unit-product = \ensuremath { { } \cdot { } } } \$\unit{\metre\second} = \text{metre second}\$
ms = millisecond	\sisetup{inter-unit-product = \ensuremath { { } \cdot { } } } \$\unit{\milli\second} = \text{millisecond}\$

在某些情况下，省略空格是常见的做法。典型的例子是 kWh，其中 “kWh” 没有添加任何有用的信息。如果重复使用这样的单位，建议 siunitx 用户创建自定义单位以确保一致性 (consistency)。需要注意的是，虽然这是常见的做法，但 bipsm 不允许这样做 [1]。

由其他单位按除法 (division) 构成的单位的符号用斜杠 (virgule)(斜线 [oblique

stroke]、正斜杠 [slash]、“/” )<sup>7</sup>、水平线或负指数 (negative exponents) 表示。然而，为了避免歧义，除非使用括号 (parentheses)，否则斜杠 (virgule) 不得在同一行上重复。当在 siunitx 中使用已命名的单位宏 (named unit macros) 时，可以确保这一点，这将“捕获 (trap)”重复除法 (repeated division) 并正确格式化它。在复杂的情况下，负指数比其他格式更受欢迎。

排版的结果	输入的命令
$\text{J mol}^{-1} \text{K}^{-1}$	<code>\unit{\joule\per\mole\per\kelvin}</code>
$\frac{\text{J}}{\text{mol K}}$	<code>\unit[per-mode = fraction]{\joule\per\mole\per\kelvin}</code>
$\text{J}/(\text{mol K})$	<code>\unit[per-mode = symbol]{\joule\per\mole\per\kelvin}</code>

乘积 (products) 和错误 (errors) 应显示哪个单位适用于每个给定的数字 (number)。因此， $(2 \times 3) \text{ m}$  是几何领域 (geometric area) 长度的有序集合 (ordered set of lengths)，而  $2 \times 3 \text{ m}$  是长度 (等于  $6 \text{ m}$ )。因此， $\times$  不是乘积，而是数学运算符 (mathematical operator)；同样， $2 \times 3$  矩阵 (matrix) 不是  $6$  矩阵！在某些区域，面积 (areas) 和体积 (volumes) 用单独的单位 (separated units) 表示，但单位提高到适当的幂 (power)： $2 \times 3 \text{ m}^2$ 。虽然这确实显示了正确的整体单位 (overall units)，但它可能会造成混淆，从而不鼓励使用。

书写数字范围 (ranges of numbers) 时必须小心。对于纯数值 (purely numerical values)，通常使用连接号 (en-dash) 来显示范围，例如“参见第 1–5 页”。另一方面，如果这样写，物理量 (physical quantities) 可能会被误解为负值 (negative values)。由于数量 (quantity) 是一个单一的数学实体 (single mathematical entity)，因此用一个连接号 (en-dash) 后跟一个单位来写数值也是不正确的。因此，强烈建议使用“to”一词。

排版的结果	输入的命令
$1 \text{ m to } 5 \text{ m long}$	<code>\qtyrange{1}{5}{\metre} long</code>

### 10.3 图形和表格

在图形 (graphs) 和表格 (tables) 中，每个条目 (entry) 或轴标记 (axis mark) 后面的单位重复 (repetition) 是混淆和重复的。因此，最好将单位放置在信息的标签部分 (label

<sup>7</sup>请注意，斜杠 (virgule) 和斜线分隔符号 (solidus) 不是同一个符号。

part)。将单位放在方括号 (square brackets) 中很常见，但数学上很差<sup>8</sup>。更好的方法是显示所有数量 (all quantities) 除以单位的情况，这会使条目 (entries) 成为无单位比率 (unitless ratios)。表 43 和图 1 对此进行了说明。

```
\begin{table}[H]
\caption{\ {\Heiti 表格标签示例}\label{tab:xmpl:unitless}}
\sisetup{
    table-number-alignment = center,
    table-format = 1.4
}
\begin{tabular}{@{}cS@{}}
\hline{1.2pt}
    {\Heiti 条目} & {\{\Heiti 长度}/\unit{\metre}} \\
\hline{0.7pt}
    1 & 1.1234 \\
    2 & 1.1425 \\
    3 & 1.7578 \\
    4 & 1.9560 \\
\hline{1.2pt}
\end{tabular}
\end{table}
```

表 43: 表格标签示例

条目	长度/m
1	1.1234
2	1.1425
3	1.7578
4	1.9560

```
\begin{figure}[H]
\begin{tikzpicture}
\begin{axis}[
```

<sup>8</sup>例如，对于加速度  $a$ ，表达式  $[a]$  是  $a$  的尺寸 (dimensions)，即在这种情况下每时间平方的长度。

```

xlabel =  $t/\text{unit}\{\text{second}\}$ ,
xmax   = 6,
xmin   = 0,
ylabel =  $d/\text{unit}\{\text{metre}\}$ ,
ymin   = 0
]
\addplot[smooth,mark=*]
plot coordinates {
(0,0)
(1,5)
(2,8)
(3,9)
(4,8)
(5,5)
(6,0)
};
\end{axis}
\end{tikzpicture}
\caption{\ {\Heiti 图形标签}(graph labelling)\ {\Heiti 示例}\label{fig:xmpl:unitless}}
\end{figure}

```

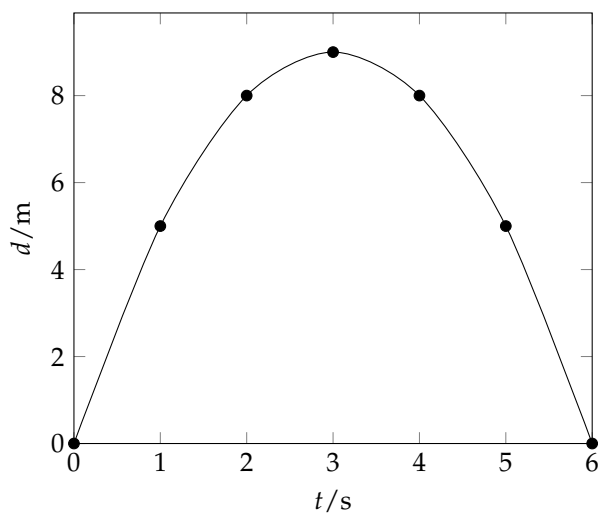


图 1: 图形标签 (graph labelling) 示例

在大多数情况下，在表体 (body of a table) 中添加指数值 (exponent values) 不如在

列表头 (column headers) 中添加固定指数 (fixed exponent) 更可取, 表 44 显示了一个示例。由于 “<”, 这里需要使用 `\multicolumn`, 如果没有 `\multicolumn`, 标题 (titles) 后面就跟着 “kg”!

```
\begin{table}[H]
  \caption{\ \Heiti 糟糕的列和好的列\label{tab:good}}
  \sisetup{table-number-alignment = center}
  \begin{tabular}{c}
    @{}
    c
    S[table-format = 1.3e1]
    @{\,\unit{\kilogram}}
    S[table-format = 2.2]
    @{}
  \end{tabular}
  \hline{1.2pt}
  {\Heiti 条目} & \multicolumn{1}{c}{\Heiti 质量} &
    {\Heiti 质量}/\qty[print-unity-mantissa = false]{e3}{\kilogram} \\
  \hline{0.7pt}
  1 & 4.56e3 & 4.56 \\
  2 & 2.40e3 & 2.40 \\
  3 & 1.345e4 & 13.45 \\
  4 & 4.5e2 & 0.45 \\
  \hline{1.2pt}
\end{tabular}
\end{table}
```

表 44: 糟糕的列和好的列

条目	质量	质量/10 <sup>3</sup> kg
1	4.56 × 10 <sup>3</sup> kg	4.56
2	2.40 × 10 <sup>3</sup> kg	2.40
3	1.345 × 10 <sup>4</sup> kg	13.45
4	4.5 × 10 <sup>2</sup> kg	0.45

## 11 安装

对于大多数用户来说，不需要显式安装 (explicitly install) siunitx：它可以从当前 T<sub>E</sub>X Live 和 MiK<sub>T</sub>E<sub>X</sub> 系统中的宏包管理系统 (package management system) 中获得。

对于手动安装 (manual installation)，该宏包可从 [ctan](#) 获得。除了原始源文件 (raw source files) 外，ctan 还将包保存为预解压缩 (pre-extracted) 的 zip 文件 `siunitx.tds.zip`。后者对大多数用户来说最方便：只需将其解压缩到本地 `texmf` 目录中即可。

该宏包需要 l3kernel 和 l3packages 捆绑包 (bundles) 中提供的 L<sup>A</sup>T<sub>E</sub>X3 的支持。这两个都包含在 T<sub>E</sub>X Live 和 MiK<sub>T</sub>E<sub>X</sub> 中，或者也可以从 ctan 以即时安装的形式 (ready-to-install form) 提供。

## 12 致谢

许多用户都为 siunitx 的新功能 (new features) 提供了反馈 (feedback)、错误报告 (bug reports) 和想法 (ideas)：感谢你们。特别感谢斯特凡·平诺 (Stefan Pinnow)，他担任 siunitx 的 beta 测试人员，发现了错误的输出、错误的文档以及文档中奇怪的拼写错误。还要感谢恩里科·格雷戈里奥 (Enrico Gregorio) 鼓励我完成一个完全兼容 expl3 的宏包版本。还要感谢丹尼·埃尔斯 (Danie Els) 和马塞尔·海尔多恩 (Marcel Heldoorn) 分别提供了 Slstyle 宏包和 Slunits 宏包，这是 siunitx 宏包开发的起点。

## 13 提出建议并报告错误

欢迎对 siunitx 的反馈，无论是提出建议还是报告问题。在发送反馈时，如果包含一个小的示例文件，显示正在报告的错误或说明所需的输出，总是很有用的。如果包含一个“参考呈现 (reference rendering)”，显示输出应该是什么样子，那么会很有帮助。典型的示例文件可能会读取

```
\listfiles
% 除非问题依赖于类，否则使用 article 类
\documentclass{article}
```

```

\usepackage{siunitx}
% 根据需要加载的其他宏包
\begin{document}
Reference output:  $1.23\,\mathrm{m}$ 
\textsf{siunitx} output:  $\qty{1.23}{\metre}$ 
\end{document}

```

如图所示，通常最好使用 `article` 类，只加载显示问题所需的宏包。在报告错误时，包含一份由  $\text{\LaTeX}$  生成的日志文件副本也很有用 (因为宏包的版本对解决问题很重要)。

反馈可以通过多种方式发送。开发代码 (development code) 和问题跟踪器 (issue tracker) 托管在 GitHub: <https://github.com/josephwright/siunitx/> 中。在那里打开的问题对其他用户是可见的，并确保它们不会被遗忘。

## 参考文献

- [1] *The International System of Units (si)*, <https://www.bipm.org/en/measurement-units/>.
- [2] *International System of Units from nist*, <http://physics.nist.gov/cuu/Units/index.html>.

# 索引

斜体数字 (*italicnumbers*) 表示描述相应条目所在的页面 (pages), 带有下列线的数字 (*numbersunderlined*) 表示定义 (definition), 所有其他数字表示使用的位置 (places)。

符号	
\!	58, 60
\,	41, 58, 63
\_	41
<b>A</b>	
\A	19
allow-quantity-breaks (option)	63
\ampere	13
\ang	6, 9, 53
\angle	51
angle-mode (option)	54
angle-separator (option)	55
angle-symbol-degree (option)	55
angle-symbol-minute (option)	55
angle-symbol-over-decimal (option)	56
angle-symbol-second (option)	55
\angstrom	82
\approx	33
\arcminute	15, 54, 55
\arcsecond	15, 54, 55
\as	18
\astronomicalunit	15
\atomicmassunit	83
\atto	16
<b>B</b>	
\bar	82
\barn	82
\becquerel	14
\bel	15
\bfseries	97
\bit	22
\bohr	82
bracket-ambiguous-numbers (option)	45
bracket-negative-numbers (option)	45
bracket-unit-denominator (option)	59
\byte	22
<b>C</b>	
\C	21
\cancel	16, 17
\candela	13
\centi	16
\clight	82
\cm	18
color (option)	32
\color	24
complex-angle-unit (option)	53
complex-mode (option)	52
complex-root-position (option)	52
complex-symbol-angle (option)	53
complex-symbol-degree (option)	53
\complexnum	6, 12, 81
\complexqty	6, 12
\coulomb	14
\cubed	13
\cubic	13

<b>D</b>		extract-mass-in-kilograms (option) . . . 64
<code>\dalton</code> . . . . .	15	<code>\extras&lt;language&gt;</code> . . . . . 98
<code>\day</code> . . . . .	11, 15	
<code>\dB</code> . . . . .	22	<b>F</b>
<code>\deca</code> . . . . .	13, 16	<code>\F</code> . . . . . 21
<code>\deci</code> . . . . .	16	<code>\farad</code> . . . . . 14
<code>\decibel</code> . . . . .	15	<code>\femto</code> . . . . . 16
<code>\DeclareSIPower</code> . . . . .	23	<code>\fF</code> . . . . . 21
<code>\DeclareSIPrefix</code> . . . . .	23	<code>\fg</code> . . . . . 17
<code>\DeclareSIQualifier</code> . . . . .	24	<code>\fH</code> . . . . . 21
<code>\DeclareSIUnit</code> . . . . .	23, 57, 94	fill-angle-degrees (option) . . . . . 55
<code>\degree</code> . . . . .	11, 15, 23, 51, 54, 55	fill-angle-minutes (option) . . . . . 55
<code>\degreeCelsius</code> . . . . .	14	fill-angle-seconds (option) . . . . . 55
<code>\deka</code> . . . . .	13	fixed-exponent (option) . . . . . 37
digit-group-first-size (option) . . . . .	43	<code>\fmol</code> . . . . . 18
digit-group-other-size (option) . . . . .	43	forbid-literal-units (option) . . . . . 62
digit-group-size (option) . . . . .	43	<code>\frac</code> . . . . . 58
display-per-mode (option) . . . . .	60	fraction-command (option) . . . . . 58
<code>\dm</code> . . . . .	18	free-standing-units (option) . . . . . 57
drop-exponent (option) . . . . .	37, 75	<code>\fs</code> . . . . . 18
drop-uncertainty (option) . . . . .	37	
drop-zero-decimal (option) . . . . .	40	<b>G</b>
<code>\DTLshowtable</code> . . . . .	89	<code>\g</code> . . . . . 18
<b>E</b>		<code>\ge</code> . . . . . 33
<code>\electronmass</code> . . . . .	82	<code>\geq</code> . . . . . 33
<code>\electronvolt</code> . . . . .	15	<code>\GetTranslation</code> . . . . . 84
<code>\elementarycharge</code> . . . . .	82	<code>\GeV</code> . . . . . 21
<code>\ensuremath</code> . . . . .	86	<code>\gg</code> . . . . . 33
<code>\eV</code> . . . . .	21	<code>\GHz</code> . . . . . 19
evaluate-expression (option) . . . . .	35	<code>\gibi</code> . . . . . 22
<code>\exa</code> . . . . .	16	<code>\giga</code> . . . . . 16
<code>\exbi</code> . . . . .	22	<code>\GPa</code> . . . . . 20
exponent-base (option) . . . . .	43	<code>\gray</code> . . . . . 14
exponent-mode (option) . . . . .	37	group-digits (option) . . . . . 42
exponent-product (option) . . . . .	43	group-minimum-digits (option) . . . . . 42
expression (option) . . . . .	35	group-separator (option) . . . . . 42
		<code>\GW</code> . . . . . 20

<b>H</b>		<code>\kHz</code> . . . . .	19
<code>\H</code> . . . . .	21	<code>\kibi</code> . . . . .	22
<code>\hartree</code> . . . . .	82	<code>\kilo</code> . . . . .	16, 23, 64, 94
<code>\hectare</code> . . . . .	15	<code>\kilogram</code> . . . . .	13
<code>\hecto</code> . . . . .	16	<code>\kJ</code> . . . . .	21
<code>\henry</code> . . . . .	14	<code>\km</code> . . . . .	18
<code>\hertz</code> . . . . .	14	<code>\kmol</code> . . . . .	19
<code>\highlight</code> . . . . .	16, 17	<code>\kN</code> . . . . .	20
<code>\hL</code> . . . . .	19	<code>\knot</code> . . . . .	82
<code>\hl</code> . . . . .	19	<code>\kohm</code> . . . . .	20
<code>\hour</code> . . . . .	15, 94	<code>\kPa</code> . . . . .	20
<code>\Hz</code> . . . . .	19	<code>\kV</code> . . . . .	20
<b>I</b>		<code>\kW</code> . . . . .	20
<code>inline-per-mode (option)</code> . . . . .	60	<code>\kWh</code> . . . . .	21, 94
<code>input-close-uncertainty (option)</code> . . . . .	34	<b>L</b>	
<code>input-comparators (option)</code> . . . . .	33	<code>\L</code> . . . . .	19
<code>input-complex-root (option)</code> . . . . .	52	<code>\l</code> . . . . .	19
<code>input-decimal-markers (option)</code> . . . . .	33	<code>\le</code> . . . . .	33
<code>input-digits (option)</code> . . . . .	33	<code>\leq</code> . . . . .	33
<code>input-exponent-markers (option)</code> . . . . .	33	<code>list-exponents (option)</code> . . . . .	49
<code>input-ignore (option)</code> . . . . .	33	<code>list-final-separator (option)</code> . . . . .	47
<code>input-open-uncertainty (option)</code> . . . . .	34	<code>list-pair-separator (option)</code> . . . . .	47
<code>input-signs (option)</code> . . . . .	33	<code>list-separator (option)</code> . . . . .	47
<code>input-uncertainty-signs (option)</code> . . . . .	34	<code>list-units (option)</code> . . . . .	50
<code>inter-unit-product (option)</code> . . . . .	58	<code>\liter</code> . . . . .	15, 86
<b>J</b>		<code>\litre</code> . . . . .	15, 86
<code>\J</code> . . . . .	20	<code>\ll</code> . . . . .	33
<code>\joule</code> . . . . .	14	<code>locale (option)</code> . . . . .	79
<b>K</b>		<code>\lumen</code> . . . . .	14
<code>\K</code> . . . . .	22	<code>\lux</code> . . . . .	14
<code>\kA</code> . . . . .	19	<b>M</b>	
<code>\katal</code> . . . . .	14	<code>\m</code> . . . . .	18
<code>\kelvin</code> . . . . .	13	<code>\mA</code> . . . . .	19
<code>\keV</code> . . . . .	21	<code>\mathnormal</code> . . . . .	86
<code>\kg</code> . . . . .	18	<code>\mathrm</code> . . . . .	58, 86

<code>\mC</code> .....	22	<code>\mW</code> .....	20
<code>\mebi</code> .....	22		
<code>\mega</code> .....	16		
<code>\meter</code> .....	12, 86		
<code>\metre</code> .....	12, 13, 86		
<code>\MeV</code> .....	21		
<code>\meV</code> .....	21		
<code>\mg</code> .....	17		
<code>\mH</code> .....	21		
<code>\MHz</code> .....	19		
<code>\mHz</code> .....	19		
<code>\micro</code> .....	16, 81		
<code>\milli</code> .....	16		
minimum-decimal-digits (option) .....	40		
minimum-integer-digits (option) .....	40		
<code>\minute</code> .....	15		
<code>\mJ</code> .....	21		
<code>\mL</code> .....	19		
<code>\ml</code> .....	19		
<code>\mm</code> .....	18		
<code>\mmHg</code> .....	82		
<code>\mmol</code> .....	18		
<code>\MN</code> .....	20		
<code>\mN</code> .....	19		
mode (option) .....	28		
<code>\Mohm</code> .....	20		
<code>\mohm</code> .....	20		
<code>\mol</code> .....	18		
<code>\mole</code> .....	13		
<code>\mp</code> .....	33		
<code>\MPa</code> .....	20		
<code>\ms</code> .....	18		
<code>\multicolumn</code> .....	26, 106		
<code>\multirow</code> .....	26		
<code>\mV</code> .....	20		
<code>\MW</code> .....	20		
		<b>N</b>	
		<code>\N</code> .....	19
		<code>\nA</code> .....	19
		<code>\nano</code> .....	16
		<code>\nauticalmile</code> .....	82
		<code>\nC</code> .....	21
		negative-color (option) .....	45
		<code>\neper</code> .....	15
		<code>\newton</code> .....	14
		<code>\nF</code> .....	21
		<code>\ng</code> .....	17
		<code>\nH</code> .....	21
		<code>\nm</code> .....	18
		<code>\nmol</code> .....	18
		<code>\ns</code> .....	18
		<code>\num</code> .....	6, 8–12
		number-angle-product (option) .....	54
		number-color (option) .....	32
		number-mode (option) .....	28
		<code>\numlist</code> .....	6, 8, 11
		<code>\numproduct</code> .....	6, 9, 11, 81
		<code>\numrange</code> .....	6, 9, 12
		<code>\nV</code> .....	20
		<code>\nW</code> .....	20
		<b>O</b>	
		<code>\of</code> .....	16
		<code>\ohm</code> .....	14
		options:	
		allow-quantity-breaks .....	63
		angle-mode .....	54
		angle-separator .....	55
		angle-symbol-degree .....	55
		angle-symbol-minute .....	55
		angle-symbol-over-decimal .....	56

angle-symbol-second .....	55	input-complex-root .....	52
bracket-ambiguous-numbers .....	45	input-decimal-markers .....	33
bracket-negative-numbers .....	45	input-digits .....	33
bracket-unit-denominator .....	59	input-exponent-markers .....	33
color .....	32	input-ignore .....	33
complex-angle-unit .....	53	input-open-uncertainty .....	34
complex-mode .....	52	input-signs .....	33
complex-root-position .....	52	input-uncertainty-signs .....	34
complex-symbol-angle .....	53	inter-unit-product .....	58
complex-symbol-degree .....	53	list-exponents .....	49
digit-group-first-size .....	43	list-final-separator .....	47
digit-group-other-size .....	43	list-pair-separator .....	47
digit-group-size .....	43	list-separator .....	47
display-per-mode .....	60	list-units .....	50
drop-exponent .....	37, 75	locale .....	79
drop-uncertainty .....	37	minimum-decimal-digits .....	40
drop-zero-decimal .....	40	minimum-integer-digits .....	40
evaluate-expression .....	35	mode .....	28
exponent-base .....	43	negative-color .....	45
exponent-mode .....	37	number-angle-product .....	54
exponent-product .....	43	number-color .....	32
expression .....	35	number-mode .....	28
extract-mass-in-kilograms .....	64	output-close-uncertainty .....	44
fill-angle-degrees .....	55	output-complex-root .....	52
fill-angle-minutes .....	55	output-decimal-marker .....	43
fill-angle-seconds .....	55	output-exponent-marker .....	43
fixed-exponent .....	37	output-open-uncertainty .....	44
forbid-literal-units .....	62	overwrite-functions .....	57
fraction-command .....	58	parse-numbers .....	35, 74
free-standing-units .....	57	parse-units .....	62
group-digits .....	42	per-mode .....	58
group-minimum-digits .....	42	per-symbol .....	59
group-separator .....	42	per-symbol-script-correction ....	60
inline-per-mode .....	60	power-half-as-sqrt .....	62
input-close-uncertainty .....	34	prefix-mode .....	64
input-comparators .....	33	print-complex-unity .....	53

print-implicit-plus .....	46	table-align-uncertainty .....	70
print-unity-mantissa .....	46	table-alignment .....	78, 79
print-zero-exponent .....	46	table-alignment-mode .....	65
print-zero-integer .....	46	table-auto-round .....	73
product-exponents .....	49	table-column-type .....	79
product-mode .....	48	table-column-width .....	76
product-phrase .....	48	table-fixed-width .....	76
product-symbol .....	48	table-format .....	68
product-units .....	50	table-number-alignment .....	65
propagate-math-font .....	30	table-text-alignment .....	78
qualifier-mode .....	61	text-family-to-math .....	30
qualifier-phrase .....	61	text-font-command .....	31
quantity-product .....	63	text-series-to-math .....	30
range-exponents .....	49	text-subscript-command .....	31
range-phrase .....	48	text-superscript-command .....	31
range-units .....	50	tight-spacing .....	46
reset-math-version .....	30	uncertainty-descriptor-mode .....	44
reset-text-family .....	29	uncertainty-descriptor-separator .....	44
reset-text-series .....	29	uncertainty-descriptors .....	44
reset-text-shape .....	29	uncertainty-mode .....	43
retain-explicit-decimal-marker ..	35	uncertainty-separator .....	44
retain-explicit-plus .....	35	unit-color .....	32
retain-negative-zero .....	36	unit-font-command .....	62
retain-zero-uncertainty .....	35	unit-mode .....	28
round-half .....	39	unit-optional-argument .....	57
round-minimum .....	39	use-xspace .....	57
round-mode .....	38	zero-decimal-as-symbol .....	46
round-pad .....	39	zero-symbol .....	46
round-precision .....	38	output-close-uncertainty (option) ....	44
round-zero-positive .....	40	output-complex-root (option) .....	52
separate-uncertainty-units .....	64	output-decimal-marker (option) .....	43
space-before-unit .....	57	output-exponent-marker (option) .....	43
sticky-per .....	61	output-open-uncertainty (option) .....	44
table-align-exponent .....	70	overwrite-functions (option) .....	57
table-align-text-after .....	72		
table-align-text-before .....	72		

## P

\Pa .....	20
-----------	----

<code>\pA</code> .....	19	<code>\qty...</code> .....	80
<code>parse-numbers (option)</code> .....	35, 74	<code>\qtylist</code> .....	6, 11, 50
<code>parse-units (option)</code> .....	62	<code>\qtyproduct</code> .....	6, 11, 50, 81
<code>\pascal</code> .....	14	<code>\qtyrange</code> .....	6, 12, 50
<code>\pebi</code> .....	22	<code>qualifier-mode (option)</code> .....	61
<code>\per</code> .....	10, 15, 23, 58–61, 95	<code>qualifier-phrase (option)</code> .....	61
<code>per-mode (option)</code> .....	58	<code>\quantity</code> .....	85
<code>per-symbol (option)</code> .....	59	<code>quantity-product (option)</code> .....	63
<code>per-symbol-script-correction (option)</code> .....	60		
<code>\percent</code> .....	13	<b>R</b>	
<code>\peta</code> .....	16	<code>\radian</code> .....	14
<code>\pF</code> .....	21	<code>\raiseto</code> .....	13
<code>\pg</code> .....	17	<code>range-exponents (option)</code> .....	49
<code>\pgfkeys</code> .....	81	<code>range-phrase (option)</code> .....	48
<code>\pH</code> .....	21	<code>range-units (option)</code> .....	50
<code>\pico</code> .....	16	<code>reset-math-version (option)</code> .....	30
<code>\planckbar</code> .....	82	<code>reset-text-family (option)</code> .....	29
<code>\pm</code> .....	17, 18, 33, 44	<code>reset-text-series (option)</code> .....	29
<code>\pmol</code> .....	18	<code>reset-text-shape (option)</code> .....	29
<code>power-half-as-sqrt (option)</code> .....	62	<code>retain-explicit-decimal-marker (option)</code> .....	35
<code>prefix-mode (option)</code> .....	64	<code>retain-explicit-plus (option)</code> .....	35
<code>print-complex-unity (option)</code> .....	53	<code>retain-negative-zero (option)</code> .....	36
<code>print-implicit-plus (option)</code> .....	46	<code>retain-zero-uncertainty (option)</code> .....	35
<code>print-unity-mantissa (option)</code> .....	46	<code>round-half (option)</code> .....	39
<code>print-zero-exponent (option)</code> .....	46	<code>round-minimum (option)</code> .....	39
<code>print-zero-integer (option)</code> .....	46	<code>round-mode (option)</code> .....	38
<code>product-exponents (option)</code> .....	49	<code>round-pad (option)</code> .....	39
<code>product-mode (option)</code> .....	48	<code>round-precision (option)</code> .....	38
<code>product-phrase (option)</code> .....	48	<code>round-zero-positive (option)</code> .....	40
<code>product-symbol (option)</code> .....	48	<code>\rowcolor</code> .....	97
<code>product-units (option)</code> .....	50		
<code>propagate-math-font (option)</code> .....	30	<b>S</b>	
<code>\ps</code> .....	18	<code>\s</code> .....	18
<code>\pV</code> .....	20	<code>\second</code> .....	13
		<code>\SendSettingsToPgf</code> .....	81
<b>Q</b>		<code>separate-uncertainty-units (option)</code> ..	64
<code>\qty</code> .....	6, 11, 12, 17, 23, 54, 56, 57, 85, 99		



<code>\uW</code> .....	20	<b>Y</b>	
		<code>\yobi</code> .....	22
<b>V</b>		<code>\yocto</code> .....	16
<code>\V</code> .....	20	<code>\yotta</code> .....	16
<code>\volt</code> .....	14	<b>Z</b>	
		<code>\zebi</code> .....	22
<b>W</b>		<code>\zepto</code> .....	16
<code>\W</code> .....	20	zero-decimal-as-symbol (option) .....	46
<code>\watt</code> .....	14, 94	zero-symbol (option) .....	46
<code>\weber</code> .....	14	<code>\zetta</code> .....	16