

Physikalische Einheiten in \LaTeX mit dem Paket `siunitx`

Alfred H. Gitter und Richard L. Gitter

21. April 2025

Inhaltsverzeichnis

1	Maßeinheiten und <code>siunitx</code>	2
1.1	Physikalische Einheiten	2
1.2	Das Paket <code>siunitx</code>	2
1.3	Einbindung und Konfiguration	3
1.4	Paket <code>units</code> als Alternative zu <code>siunitx</code>	3
2	Zahlen und Winkel	4
2.1	Zahlen bilden mit <code>\num{...}</code>	4
2.2	Optionen für Zahlen mit <code>\num[...]{...}</code>	4
2.3	Listen und Tabellen mit Zahlen	6
2.4	Winkel	6
3	Physikalische Einheiten	7
3.1	Einheiten bilden mit <code>\unit{...}</code>	7
3.2	Gute Vorsätze	9
3.3	Optionen für Einheiten mit <code>\unit[...]{...}</code>	9
3.4	Definition zusätzlicher Maßeinheiten	10
4	Werte physikalischer Größen	11
4.1	Größenwerte bilden mit <code>\qty{...}{...}</code>	11
4.2	Optionen für Größen mit <code>\qty[...]{...}{...}</code>	11
4.3	Listen, Produkte und Bereiche von Größen	11
4.4	Notation der Messunsicherheit	12

1 Maßeinheiten und siunitx

1.1 Physikalische Einheiten

Der Wert (englisch: value) einer skalaren (das heißt: richtungsunabhängigen) physikalischen Größe (physical quantity) besteht aus einem Zahlenwert (numerical value) und einer Maßeinheit (unit of measurement, unit).

In der Regel werden in der mathematischen Notation Zahlen sowie physikalische Maßeinheiten (mit ihren Vorsätzen) steil (gerade), Unbekannte hingegen kursiv gesetzt. Letzteres trifft meistens auch auf Konstanten zu. Nach der Norm DIN EN ISO 80000-2:2013-08 sollen mathematische Konstanten und feststehende Funktionsnamen, wie \sin , allerdings steil gesetzt werden. In \LaTeX wird ein steiles Pi, also π , im Mathematikmodus (math mode) mit dem Befehl `\uppi` des Pakets *upgreek* erzeugt.

In Listen von Größenwerten mit gleicher Einheit sollte jeder Wert mit der Einheit versehen sein, wie zum Beispiel bei 1 m, 2 m und 3 m. In Produkten von Größenwerten mit gleicher Einheit sollten entweder die multiplizierten Zahlen in runden Klammern vor der Einheit des Produkts stehen, oder jeder Wert mit seiner Einheit als Faktor geschrieben werden, also zum Beispiel $(2 \cdot 5) \text{ m}^2 = 2 \text{ m} \cdot 5 \text{ m}$.

In Diagrammen und Tabellen, die Werte physikalischer Größen zeigen, sollte man nicht wiederholt die gleiche Einheit wiedergeben. Besser ist, in Achsenbeschriftungen beziehungsweise im Tabellenkopf, die Größe durch ihre Einheit zu teilen oder *Größensymbol* in *Einheit* zu schreiben, sodass Werte ohne Einheit (Zahlen) entstehen. Beispielsweise schreibt man Länge ℓ / m oder Länge ℓ in m.

In wissenschaftlicher Literatur und amtlichen Texten verwendet man hauptsächlich die Maßeinheiten des [Internationalen Einheitensystems](#) (SI), das vom [Internationalen Büro für Maß und Gewicht](#) (BIPM) bereitgestellt wird, gemäß der [Beschlüsse der Generalkonferenz für Maß und Gewicht](#). Das SI enthält sieben Basiseinheiten, sowie davon abgeleitete Einheiten. Es gibt darüber hinaus Maßeinheiten außerhalb des SI, welche veraltet oder für bestimmte Anwendungen weiterhin üblich sind.

1.2 Das Paket siunitx

Das Schreiben von SI-Maßeinheiten und damit gebildeten Werten physikalischer Größen wird in \LaTeX durch das Paket *siunitx* von [Joseph Wright](#) erleichtert, auch im Mathematikmodus (math mode). Wir betrachten hier die Version 3.4.8 vom 26.3.2025. Vorliegendes Manuskript ist lediglich eine kurze, einführende Darstellung von *siunitx*, die

nicht die ausführliche [englische Anleitung](#) ersetzt.

Man beachte, dass `siunitx` nur die Einheiten des SI bereitstellt. Maßeinheiten außerhalb des SI kann der Nutzer allerdings selbst definieren. So lassen sich alle Maßeinheiten einheitlich darstellen und mit einer Vielfalt von Optionen formatieren. Überdies erleichtert das Paket auch die richtige Schreibweise von Zahlen und Winkeln.

1.3 Einbindung und Konfiguration

Nach der Einbindung des Pakets durch

```
\usepackage{siunitx}
```

wird es mit dem Befehl `\sisetup{...}` konfiguriert, wobei `...` eine Option in der Form *Schlüssel* oder *Schlüssel = Wert* setzt. Man kann auch mehrere Optionen angeben, die durch Kommata getrennt sind. Die Anpassung an deutsche Schreibweisen durch

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

bewirkt, dass bei der Zahlenausgabe ein Komma als Dezimaltrennzeichen geschrieben wird (wie mit der Option `output-decimal-marker = {,}`). Bei der Eingabe von Dezimalzahlen dürfen aber Komma und Punkt verwendet werden. Außerdem bewirkt `locale = DE`, dass in der Exponentialschreibweise von Zahlen das Zeichen `·` zwischen Mantisse und Potenz steht (wie mit der Option `exponent-product = \cdot`), und dass als Multiplikationszeichen von Einheiten ein kleines Leerzeichen (erzeugt durch `\,`) dient (wie mit der Option `inter-unit-product = {\,}`), und dass einige Wörter ins Deutsche übersetzt werden (zum Beispiel `and` → `und`).

Alle Optionen können mit `\sisetup{...}` global (im ganzen folgenden Dokument) wirken oder, eingefasst in eckigen Klammern, lokal wirken, das heißt: einen bestimmten Ausgabebefehl anpassen, siehe unten. Globale Optionen können auch schon beim Befehl `\usepackage{siunitx}` angegeben werden (in eckigen Klammern).

1.4 Paket `units` als Alternative zu `siunitx`

Wenn man nur gelegentlich Werte mit Einheiten schreibt, ist das relativ einfache Paket [units](#) von Axel Reichert nützlich. Es bietet, neben dem Befehl `nicefrac`, zwei Befehle zur Formatierung von Einheiten oder Werten mit Einheiten, auch als Bruch. Man kann sie in normalem Text oder im Mathematikmodus (`math mode`) verwenden:

$\$ \backslash \text{unit}\{N\} = \backslash \text{unit}\{\text{kg}\} \backslash \text{cdot} \backslash \text{unit}\{\text{m}/\text{s}^2\} \$$
 $\$ v \$$ war $\backslash \text{unitfrac}[2\{,\}5]\{\text{m}\}\{\text{s}\}$

$N = \text{kg} \cdot \text{m}/\text{s}^2$
 v war $2,5 \text{ m}/\text{s}$

Die Befehle können Einheiten oder Zahlen mit Einheiten wiedergeben. Ein steiles (nicht kursives) μ erhält man mit dem Befehl `\textmu`; ebenso ein Ω mit `\textohm`.

Die Pakete *siunitx* und *units* sind leider nicht kompatibel, das heißt: man kann nur entweder das eine oder das andere Paket verwenden. Allerdings ist das Paket *nicefrac*, das lediglich den Befehl `\nicefrac` bereitstellt, kompatibel zu *siunitx*.

2 Zahlen und Winkel

2.1 Zahlen bilden mit `\num{...}`

Zahlen werden mit `\num{...}` ausgegeben, wobei gegebenenfalls ein Dezimalkomma richtig (ohne anschließenden Leerraum) gesetzt wird. In der Eingabe von Zehnerpotenzen wird der Exponent durch ein vorangestelltes e, E, d oder D gekennzeichnet:

<code>\num{12345.678}</code> %(Eingabe mit Dezimalpunkt)	12 345,678
<code>\num{12345,678}</code> %(Eingabe mit Dezimalkomma)	12 345,678
<code>\num{-1234,5 e 6}</code> %(vor/nach e Leerzeichen erlaubt)	$-1234,5 \cdot 10^6$

wobei hier `locale = DE` als globale Option zur Ausgabe verwendet wurde. Zur besseren Lesbarkeit großer Zahlen (mindestens fünf Ziffern) werden Dreierblöcke von Ziffern normalerweise durch ein vorangestelltes schmales Leerzeichen abgesetzt. Dieses Trennzeichen kann man in einer Option ändern (mit dem Schlüssel `group-separator`).

Eine komplexe Zahl kann man durch `\complexnum{...}` ausgegeben. Bei der Eingabe ist *i* oder *j* die imaginäre Einheit, bei der Ausgabe normalerweise ein nachgestelltes *i*.

<code>\complexnum{1,5+2,5i}</code>	$1,5 + 2,5i$
------------------------------------	--------------

2.2 Optionen für Zahlen mit `\num[...]{...}`

Optionen in eckigen Klammern ändern das Ausgabeformat. Ohne Optionen lässt man die eckigen Klammern weg. Mit einer entsprechenden Option können negative Zahlen rot oder positive Zahlen mit vorangestelltem Pluszeichen gedruckt werden. Man kann auch eine Zahl farbig wiedergeben. Das Paket *xcolor* stellt mehr Farben bereit.

<code>\num[negative-color = red]{-123,45}</code>	$-123,45$
<code>\num[retain-explicit-plus]{+123,45}</code>	$+123,45$
<code>\num[number-color = blue]{+123,45}</code>	$123,45$

Mit der Option `exponent-mode = threshold` werden Zahlen, deren dekadischer Logarithmus zwischen einem Minimal- und einem Maximalwert liegt, automatisch in Exponentialschreibweise gesetzt. Minimal- und Maximalwert können mit dem Schlüssel `exponent-thresholds` verändert werden. Standardmäßig ist der Minimalwert -3 und der Maximalwert 3 .

<code>\num[exponent-mode = threshold]{1234,5}</code>	$1,2345 \cdot 10^3$
<code>\num[exponent-mode = threshold, % exponent-thresholds = -4:4]{1234,5}</code>	$1234,5$

Man kann Zahlen runden lassen, indem man die Anzahl der Nachkommastellen begrenzt (`round-mode = places`) und festlegt, zum Beispiel auf 1 (`round-precision = 1`).

<code>\num[round-mode = places,round-precision = 1]{123,45}</code>	$123,5$
--	---------

Will man die Zahl π als Symbol verwenden, muss man eine entsprechende Option (global oder lokal) setzen. Damit sind aber zum Beispiel Rundungen nicht mehr möglich.

<code>\num[input-digits = 0123456789\pi]{2\pi}</code>	2π
---	--------

Auch für die Darstellung komplexer Zahlen gibt es Optionen:

Mit der Option `[complex-root-position = before-number]` kann man die imaginäre Einheit vor den Imaginärteil schreiben. Die Option `[output-complex-root = j]` bestimmt ein kursives j als imaginäre Einheit.

<code>\complexnum[complex-root-position = before-number]{3+j2}</code>	$3 + i2$
<code>\complexnum[output-complex-root = j]{3+j2}</code>	$3 + 2j$

Die folgenden Befehle ergeben die Ausgabe eines vorangestellten, aufrechten i als imaginäre Einheit mit einer kleinen Lücke vor dem Imaginärteil. Der erste Befehl sollte als globale Option schon in der Präambel stehen, nach der Einbindung des Pakets `siunitx`.

<code>\sisetup{complex-root-position = before-number, output-complex-root = \ensuremath{\mathrm{i}\,},}</code>	$1 + i2$
<code>\complexnum{1+2i}</code>	

2.3 Listen und Tabellen mit Zahlen

Der Befehl `\numlist[...]{...}` gibt eine Liste von Zahlen aus, wobei die Elemente in der Eingabe durch Semikola getrennt werden. Zur Ausgabe eines Zahlenprodukts mit dem Befehl `\numproduct{...}` werden die Zahlen in der Eingabe durch ein `x` getrennt. Ein Zahlenbereich kann mit dem Befehl `\numrange{...}` ausgegeben werden.

<code>\numlist{1; 2; 3}</code>	1, 2 und 3
<code>\numproduct{1 x 2 x 3}</code>	$1 \times 2 \times 3$
<code>\numproduct[product-symbol = \ensuremath{\cdot}]{7x7}</code>	$7 \cdot 7$
<code>\numrange[range-phrase = \,--\,]{1}{10}</code>	1–10

Für Tabellen mit Zahlenspalten stellt das Paket *siunitx* in der *tabular*-Umgebung den neuen Spaltentyp `S` bereit. Hier ein Beispiel, welches das Paket *booktabs* benötigt:

<pre>\begin{tabular}{S} \toprule % (mit Paket booktabs) {Zahlen} \\ % keine Zahl, {}-Block \midrule % (mit Paket booktabs) 3.14 \\ % Gleitkommazahl 6,0224e23 \\ % wissenschaftlich \color{red} -42 \\ % mit Farbe \bottomrule % (mit Paket booktabs) \end{tabular}</pre>	<table><tr><td style="border-top: 1px solid black; border-bottom: 1px solid black; text-align: center;">Zahlen</td></tr><tr><td style="text-align: center;">3,14</td></tr><tr><td style="text-align: center;">$6,0224 \cdot 10^{23}$</td></tr><tr><td style="text-align: center; color: red;">-42</td></tr></table>	Zahlen	3,14	$6,0224 \cdot 10^{23}$	-42
Zahlen					
3,14					
$6,0224 \cdot 10^{23}$					
-42					

Das Dezimaltrennzeichen wird in der Mitte der Spalte platziert und alle Zahleneinträge werden entsprechend ausgerichtet. In der Zahlenspalte `S` braucht und soll kein `\num{}` verwendet werden.

2.4 Winkel

Winkel werden mit dem Befehl `\ang{...}` eingegeben. Dabei kann der eingegebene Wert eine einzelne Dezimalzahl sein, welche den Winkel in Grad darstellt, oder ein Tripel aus Grad, Winkelminute und Winkelsekunde, getrennt durch Semikola. In letzterem sollten die Werte von Grad und Winkelminute ganzzahlig sein.

Die Option `angle-mode = arc` erzeugt eine Ausgabe mit den Einheiten Grad $^\circ$, Winkelminute $'$ und Winkelsekunde $''$, wobei ganzzahlige Werte von Grad und Winkelminute ausgegeben werden. Andererseits wird der Winkel mit der Option `angle-mode = decimal` in der Einheit Grad $^\circ$ ausgegeben, wobei der Wert eine Dezimalzahl ist. Eine

Umrechnung erfolgt jedoch nur dann, wenn Dezimalzahlen mit Punkten (nicht Komma) eingegeben werden, andernfalls wird eine Fehlermeldung erzeugt. Wird keine der beiden Optionen angegeben, richtet sich das Ausgabeformat nach dem Eingabeformat.

<code>\ang{12.34}</code>	12,34°
<code>\ang[angle-mode = arc]{12.34}</code>	12°20'24"
<code>\ang{1;6;36}</code>	1°6'36"
<code>\ang[angle-mode = decimal]{1;6;36}</code>	1,11°
<code>\ang{; 3 ;}</code>	3'

Man kann zwischen Grad und Winkelminuten sowie zwischen Winkelminuten und Winkelsekunden ein kleines Leerzeichen einfügen:

`\ang[angle-separator = \,]{11; 22; 33}` 11° 22' 33"

3 Physikalische Einheiten

3.1 Einheiten bilden mit `\unit{...}`

Eine physikalische Einheit wird mit dem Befehl `\unit{...}` gebildet. Das Pflichtargument kann eine Zeichenkette ohne Befehle (die durch `\` eingeleitet werden) enthalten:

die Einheit `\unit{s}` für Zeitangaben die Einheit s für Zeitangaben

Die Multiplikation von Einheiten, die derart ohne Befehle dargestellt werden, wird durch das Zeichen `.` oder `~`, die Potenzierung durch `^` erreicht.

`\unit{N} = \unit{kg.m.s^{-2}} = \unit{kg~m/s^2}` $N = \text{kg m s}^{-2} = \text{kg m/s}^2$

Von der Möglichkeit, Einheiten ohne besondere Befehle zu bilden, wird jedoch abgeraten, weil Eingabefehler schlechter erkannt werden und einige Methoden der Formatierung fehlen. Stattdessen sollte man Befehle zur Gestaltung der Einheiten verwenden. Eine einfache (nicht zusammengesetzte) Einheit kann mit einem einzelnen, dafür vorgesehenen Befehl erzeugt werden. In der Regel gibt es auch eine abgekürzte Variante:

die Einheit `\unit{\second}` für Zeitangaben die Einheit s für Zeitangaben
 die Einheit `\unit{s}` für Zeitangaben die Einheit s für Zeitangaben

Folgende Befehle (macros) sind für Einheiten verfügbar:

`\ampere` (Einheit der elektrischen Stromstärke), `\arcminute` (Winkelminute), `\arcsecond` (Winkelsekunde), `\astronomicalunit` (149 597 870 700 m), `\becquerel` (Einheit der

Radioaktivität), `\bel` (10 dB), `\bit` (Einheit für digitale Daten), `\byte` (8 bit), `\candela` (Einheit der Lichtstärke), `\coulomb` (Einheit der elektrischen Ladung), `\dalton` (atomare Masseneinheit), `\day` (Zeiteinheit: Tag), `\decibel` (logarithmisches Verhältnismaß), `\degree` (Winkelgrad), `\degreeCelsius` (Einheit der Temperatur), `\electronvolt` ($1,602\,176\,634 \cdot 10^{-19}$ J), `\farad` (Einheit der elektrischen Kapazität), `\gram` (10^{-3} kg), `\gray` (absorbierte Energie ionisierender Strahlung pro Masse), `\hectare` ($10\,000$ m²), `\henry` (Einheit der magnetischen Induktivität), `\hertz` (Einheit der Frequenz), `\hour` (Zeiteinheit: Stunde), `\joule` (Einheit der Energie), `\katal` (Einheit der katalytischen Aktivität), `\kelvin` (Einheit der Temperatur), `\kilogram` (Einheit der Masse), `\litre` oder `\liter` (10^{-3} m³), `\lumen` (Einheit des Lichtstroms), `\lux` (Einheit der photometrischen Beleuchtungsstärke), `\metre` oder `\meter` (Einheit der Länge), `\minute` (Zeiteinheit: Minute), `\mole` (Einheit der Stoffmenge), `\neper` (logarithmisches Verhältnismaß), `\newton` (Einheit der Kraft), `\ohm` (Einheit des elektrischen Widerstands), `\pascal` (Einheit des Drucks), `\percent` (lineares Verhältnismaß), `\radian` (Einheit des Winkels im Einheitskreis), `\second` (Zeiteinheit: Sekunde), `\siemens` (Einheit des elektrischen Leitwerts), `\sievert` (Einheit der Äquivalentdosis von ionisierender Strahlung), `\steradian` (Einheit des Raumwinkels), `\tesla` (Einheit der magnetischen Flussdichte), `\tonne` (1000 kg), `\volt` (Einheit der elektrischen Spannung), `\watt` (Einheit der Leistung), `\weber` (Einheit des magnetischen Flusses).

Sowohl `\litre` als auch `\liter` ergibt den Großbuchstaben L, der besser lesbar ist als der Kleinbuchstabe l. Will man dennoch den Kleinbuchstaben, kann man den Befehl `\DeclareSIUnit{\litre}{l}` in der Präambel einfügen. Dieser Befehl wirkt auf `\litre` und `\liter`. Auch für die Längeneinheit m gibt es zwei gleichwertige Befehle, nämlich `\metre` und `\meter`. In Prozentangaben kann, wie bei Einheiten, ein kleines Leerzeichen enthalten sein. Dies wird mit `\percent` erreicht, das wie eine Einheit verwendet wird.

Der Befehl `\of` ermöglicht es, der Unsitte zu frönen, Einheiten mit einem beschreibenden Index zu versehen, obwohl ein solcher nur an Größensymbolen stehen sollte:

`\unit{\volt\of{eff}}` V_{eff} Einheit für „effektive Spannung“ ?

Ein Produkt wird durch Hintereinanderschreiben von Einheitenbefehlen erzeugt:

`\unit{\joule} = \unit{\watt\second}` $J = W s$

Potenzen von Einheiten erhält man durch besondere Befehle vor der betreffenden Einheit, nämlich `\square`, `\cubic` und `\raiseto{...}`. Der Kehrwert einer Einheit, oder einer potenzierten Einheit, wird durch den vorangestellte Befehl `\per` erzeugt. Alternativ gibt es auch nachgestellte Befehle (`\squared`, `\cubed`, `\tothe{...}`).

`\unit{\cubic\meter}` m^3
`\unit{\watt\per\square\meter\raiseto{-4}\kelvin}` $W m^{-2} K^{-4}$
`\unit{\meter\cubed\second\tothe{-4}}` $m^3 s^{-4}$

Für viele Einheiten, auch mit Vorsätzen, gibt es abgekürzte Befehle, zum Beispiel:

`\ng, \ug, \mg, \g, \kg; \nm, \um, \mm, \cm, \dm, \m, \km; \ns; \us, \ms, \s; \umol, \mmol, \mol; \pA, \nA, \uA, \mA, \A; \ul, \ml, \l, \uL, \mL, \L; \Hz, \kHz, \MHz, \GHz; \N, \kN; \Pa; \kohm, \Mohm; \uV, \mV, \V, \kV; \uW, \mW, \W, \kW, \MW; \J, \kJ, \eV, \keV, \MeV, \kWh; \pF, \nF, \uF, \mF, \F; \H; \C; \K; \dB`

Alle Befehle für Einheiten, deren Exponenten und Vorsätze (s. nächstes Kapitel) sowie die eben beschriebenen abgekürzten Befehle sind standardmäßig nur als Argument der Befehle `\unit{...}` und `\qty{...}{...}` definiert, sodass gleichlautende Befehle anderer Pakete nicht verdrängt werden.

3.2 Gute Vorsätze

Neben noch kleineren und größeren sind folgende Einheitenvorsätze (Präfixe) definiert:

`\yocto` (10^{-24}), `\zepto` (10^{-21}), `\atto` (10^{-18}), `\femto` (10^{-15}), `\pico` (10^{-12}), `\nano` (10^{-9}), `\micro` (10^{-6}), `\milli` (10^{-3}), `\centi` (10^{-2}), `\deci` (10^{-1}), `\deca` (10^1), `\hecto` (10^2), `\kilo` (10^3), `\mega` (10^6), `\giga` (10^9), `\tera` (10^{12}), `\peta` (10^{15}), `\exa` (10^{18}), `\zetta` (10^{21}), `\yotta` (10^{24}); alternativ zu `\deca` ist auch die Schreibweise `\deka` möglich.

`\unit{\tonne} = \unit{\mega\gram}` `t = Mg`

Insbesondere für die Einheiten bit und byte gibt es Vorsätze, die Zweierpotenzen (nicht Zehnerpotenzen) bezeichnen (Binärpräfixe). Folgende Befehle sind definiert:

`\kibi` ($2^{10} = 1024 \neq 10^3$), `\mebi` ($2^{20} = 1\,048\,576 \neq 10^6$), `\gibi` ($2^{30} = 1\,073\,741\,824 \neq 10^9$), `\tebi` (2^{40}), `\pebi` (2^{50}), `\exbi` (2^{60}), `\zebi` (2^{70}), `\yobi` (2^{80})

`\unit{\giga\watt}` ist nicht `\unit{\gibi\byte}` GW ist nicht GiB
`\qty{1024}{\kibi\byte} = \qty{1}{\mebi\byte}` 1024 KiB = 1 MiB

3.3 Optionen für Einheiten mit `\unit[...]{...}`

Der Befehl `\per` wirkt normalerweise nur auf die erste folgende Einheit. Wenn man aber vorher die Option `sticky-per` setzt, wirkt `\per` auf alle folgenden Einheiten:

`\unit[sticky-per]{\per\meter\second}` $m^{-1} s^{-1}$

Einheiten, die aus einem Bruch bestehen, können unterschiedlich geschrieben werden:

<code>\unit{\kilogram\meter\per\second}</code>	kg m s^{-2}
<code>\unit[per-mode=symbol]{\micro\meter\per\day}</code>	$\mu\text{m/d}$
<code>\unit[per-mode=fraction]{\coulomb\per\mole}</code>	$\frac{\text{C}}{\text{mol}}$

Durchstreichungen sind nützlich, um das Kürzen gleicher Einheiten in Zähler und Nenner eines Bruchs zu zeigen. Man braucht dafür das Paket *cancel* von [Donald Arseneau](#).

<code>\unit[per-mode=fraction]{\cancel{m}\per\cancel{m}\per{s}}</code>	$\frac{\cancel{m}}{\cancel{m}s}$
--	----------------------------------

3.4 Definition zusätzlicher Maßeinheiten

Frühere Versionen des Pakets *siunitx* definierten mehr Befehle für Einheiten außerhalb des SI und Konstanten, zum Beispiel `\atomicmassunit` (1 Da), `\angstrom` (10^{-10} m), `\bar` (10^5 Pa), `\barn` (10^{-28} m²), `\bohr` (Radius des Wasserstoffatoms), `\cflight` (Vakuumlichtgeschwindigkeit), `\mmHg` (eine Druckeinheit, die noch in der Medizin verwendet wird), `\electronmass` (Elektronenmasse), `\elementarycharge` (Elementarladung = $1,602\,176\,634 \cdot 10^{-19}$ C), `\hartree` (atomare Energieeinheit nach Hartree), `\knot` (Geschwindigkeitsmaß in See- und Luftfahrt), `\nauticalmile` (1852 m, Längenmaß in See- und Luftfahrt) und `\planckbar` (reduzierte Planck-Konstante). Sie werden nun jedoch als veraltet (deprecated) angesehen und es gibt sie zukünftig nicht mehr im Paket. Benötigt man sie, muss man sie selbst definieren.

Eigene oder fehlende Einheiten werden mit dem Befehl `\DeclareSIUnit{...}{...}` definiert. Er sollte bereits in der Präambel stehen, nach dem Laden des Pakets *siunitx*. Beispiele sind dpt als Einheit der optischen Brechkraft, °F als Temperatureinheit, und (mit dem Paket *marvosym*) die Einheit € für den Anschaffungswiderstand.

```
% in der Präambel
\DeclareSIUnit{\dpt}{dpt}
\DeclareSIUnit{\degreeFahrenheit}{\text{\textcircled{F}}}
\DeclareSIUnit{\euro}{\text{\text{EUR}}}
% in der document-Umgebung
Brechkraft in \unit{\dpt} = \unit{\per\meter} \backslash
Temperatur in Florida: $T = \qty{85}{\degreeFahrenheit}$ \backslash
Strompreis, D. 2022, 2. HJ: \qty{0,4007}{\euro\per\kilo\watt\per\hour}
```

ergibt

```
Brechkraft in dpt = m-1
Temperatur in Florida: T = 85 °F
Strompreis, D. 2022, 2. HJ: 0,4007 € kW-1 h-1
```

Für Einheiten des angloamerikanischen Maßsystems gibt es das nicht in CTAN enthaltene Paket *siunitx* von [Mitchell Paulus](#) auf [GitHub](#). Währungen können systematisch mit dem Paket *currency* von [Antoine Lejay](#) als dreibuchstabile Abkürzung (gemäß ISO 4217) oder als Symbol gesetzt werden. Allerdings muss man die einzelnen Währungsdefinitionen selbst (in der Präambel) einfügen.

4 Werte physikalischer Größen

4.1 Größenwerte bilden mit `\qty{...}{...}`

Der Wert einer physikalischen Größe besteht aus Zahl und Einheit. Er kann mit dem Befehl `\qty{...}{...}` ausgegeben werden. In der ersten geschweiften Klammer wird die Zahl angegeben (wie beim Befehl `\num{...}`) und in der zweiten geschweiften Klammer die Einheit (wie beim Befehl `\unit{...}`).

$$\$E = \SI{3,8E6}{\mega\volt\per\meter}\$ \qquad E = 3,8 \cdot 10^6 \text{ MV m}^{-1}$$

4.2 Optionen für Größen mit `\qty[...]{...}{...}`

Durch Optionen in einer eckigen Klammer vor den geschweiften Klammern kann die Darstellung verändert werden.

Die Pflichtargumente von `\qty{...}{...}` werden normalerweise untersucht und verarbeitet (parsing). Diese Funktion kann man für das erste Pflichtargument abstellen. Dann wird der Zahlenwert ohne parsing als L^AT_EX-Code im math-Modus wiedergegeben:

$$\backslash\text{qty}[parse-numbers=false]{\sqrt{2}}{\m} \qquad \sqrt{2} \text{ m}$$

4.3 Listen, Produkte und Bereiche von Größen

Mit dem Befehl `\qtylist[...]{...}{...}` wird eine Liste von Größenwerten mit der gleichen Einheit ausgegeben, ähnlich wie bei Zahlen mit `\numlist[...]{...}` (siehe Seite 6). Entsprechend werden Größenprodukte oder -bereiche ausgegeben.

<code>\qtylist{1;2;3}{\meter}</code>	1 m, 2 m und 3 m
<code>\qtyproduct{1,23 x 4,56}{\meter}</code>	1,23 m × 4,56 m
<code>\qtyproduct[product-units = bracket-power]{1 x 2}{\m}</code>	(1 × 2) m ²
<code>\qtyrange{25}{30}{\cm}</code>	25 cm bis 30 cm

Dabei ist `\cm` die abgekürzte Variante von `\centi\meter` (siehe Abschnitt 3.1).

4.4 Notation der Messunsicherheit

Das Ergebnis einer Messung gibt den Wert einer Größe mit einer Messunsicherheit wieder. Diese hat die gleiche Einheit wie der Messwert. Sie kann, für die letzten Ziffern des Messwerts, in runden Klammern angegeben werden. Mit einer Option kann man eine Lücke zwischen Messwert und Messunsicherheitsklammer setzen.

<code>\qty{3,5(1)}{\volt}</code>	3,5(1) V
<code>\qty[uncertainty-separator={\,}]{3,5(1)}{\volt}</code>	3,5 (1) V

Alternativ kann die Ausgabe der Messunsicherheit durch das Zeichen \pm vom Wert getrennt werden, wobei die Eingabe auch mit dem Zeichenpaar $+-$ möglich ist.

<code>\qty[separate-uncertainty]{3,5(1)}{\volt}</code>	(3,5 ± 0,1) V
<code>\qty[separate-uncertainty]{3,5+-0,1}{\volt}</code>	(3,5 ± 0,1) V

Schließlich kann man die Einheit bei der Angabe der Messunsicherheit wiederholen:

`\qty[separate-uncertainty,separate-uncertainty-units = repeat]{8+-1}{\m}`

gibt $8\text{ m} \pm 1\text{ m}$ aus. Mit Klammer schreibt man die Einheit nur einmal, siehe oben.