Metro

Contents

1 Introduction

The Metro package aims to be a port of the Latex package siunitx. It allows easy typesetting of numbers and units with options. This package is very early in development and many features are missing, so any feature requests or bug reports are welcome!

Metro's name comes from Metrology, the study scientific study of measurement.

2 Usage

2.1 Options

`#metro-setup(..options)`

Options for Metro's can be modified by using the `metro-setup` function. It takes an argument sink and saves any named parameters found. The options for each function are specified in their respective sections.

All options and function parameters use the following types:

`Literal` Takes the given value directly. Input type is a string, content and sometimes a number.
`Switch` On-off switches. Input type is a boolean.
`Choice` Takes a limited number of choices, which are described separately for each option. Input type is a string.
`Number` Takes a float or integer.

2.2 Numbers

`#num(number, e: none, pm: none, ..options)`

Formats a number.

number `Number`
The number to format.

pm `Literal` (default: `none`)
Uncertainty of the number.

e `Number` (default: `none`)
Exponent. The exponent is applied to both the number and the uncertainty if given.

123
-1234
12345

2.2.1 Options

times `c`
The symbol

2.3 Units

`#unit(unit, ..options)`

Typsets a unit and provides full control over output format for the unit. This function can be used in two different ways.

The first way is "literally" in math mode, where the math content is parsed and any non-prefixes are separated by the `inter-unit-product` (by default a thin space) option and fractions are modified depending on the `frac-mode` option. Units within the `unit` function can be specified in three ways: a single letter (`$unit(m)$`, m); a string (`$unit("mol"^2)$`, mol^2); or by importing the units as variable definitions:

```
#import "@preview/metro:0.1.0": unit, kg, mol
#unit($kg m/s^2$)
$unit(g_"polymer" mol_"cat" s^(-1))$
```

$\mathrm{kg\ m/s^2}$

$\mathrm{g_{polymer}\ mol_{cat}\ s^{-1}}$

The second way is an "intrepreted" system by passing the function a string delimited by the `interpreted-delimeter` option (default is "#" to reflect Typst's equivalent of Latex's backslash).

```
#unit("#kilo#metre#cubed#second")\
#unit("#kilo#gram#metre#per#square#second")\
#unit("#gram#per#cubic#centi#metre")\
#unit("#square#volt#cubic#lumen#per#farad")\
#unit("#metre#squared#per#gray#cubic#lux")\
#unit("#henry#second")
```

$\mathrm{km^3\ s}$
$\mathrm{kg\ m\ s^{-2}}$
$\mathrm{g\ cm^{-3}}$
$\mathrm{V^2\ lm^3\ F^{-1}}$
$\mathrm{m^2\ Gy^{-1}\ lx^3}$
$\mathrm{H\ s}$

On its own, this is less convenient than the literal method. However, the package allows you to alter a variety of options.

2.3.1 Using Interpreted Mode

I'm not sure what to call these as in siunitx they are macros but here they are strings. But to not hurt my brain further trying to create a new name (apart from "thingymajigs") I'll call them macros for now.

When writing in interpreted mode several macros have been defined.

`#per` used as in "metres *per* second".

```
#unit("#metre#per#second")
```

$\mathrm{m\ s^{-1}}$

`#square` and `#cubic` apply their respective powers to the units after them, while `#squared` and `#cubed` apply to units before them.

```
#unit("#square#becquerel")\
#unit("#joule#squared#per#lumen")\
#unit("#cubic#lux#volt#tesla#cubed")
```

$\mathrm{Bq^2}$
$\mathrm{J^2\ lm^{-1}}$
$\mathrm{lx^3\ V\ T^3}$

Generic powers can be inserted using the `#tothe` and `#raiseto` macros. These act as functions and whatever is wrapped in its parantheses are taken as its argument.

```
#unit("#henry#tothe(5)")\
#unit("#raiseto(4.5)#radian")
```

H^5
$rad^{4.5}$

Generic qualifiers are available using the `#of` macro:

```
#unit("#kilogram#of(metal)")\
#unit("#milli#mole#of(cat)#per#kilogram#of(prod)", qualifier-mode: "bracket")
```

$\mathrm{kg}_{\mathrm{metal}}$
$\mathrm{mmol}(\mathrm{cat})\,\mathrm{kg}(\mathrm{prod})^{-1}$

2.3.2 Options

The following options affect both literal and interpreted mode.

inter-unit-product `Literal` (default: `sym.space.thin`)

The separator between each unit. The default setting is a thin space: another common choice is a centred dot.

```
#unit("#farad#squared#lumen#candela")\
#unit("#farad#squared#lumen#candela", inter-unit-product: $dot.c$)
```

$\mathrm{F}^2\,\mathrm{lm}\,\mathrm{cd}$
$\mathrm{F}^2 \cdot \mathrm{lm} \cdot \mathrm{cd}$

per-symbol `Literal` (default: `h(0pt) + sym.slash + h(0pt)`)

The symbol to use to separate the two parts of a unit when `per-symbol` is `"symbol"`.

```
#unit("#joule#per#mole#per#kelvin", per-mode: "symbol", per-symbol: [ div ])
```

J div (mol K)

The following option affects only literal mode.

frac-mode `Choice` (default: `"symbol"`)

Use to alter the handling of a `math.frac` function. These are normally created in math mode by using a slash `/`.

symbol Separates the numerator and the denominator using the symbol in `per-symbol`.

```
$unit(joule / \(mole kelvin))$\
$unit(meter / second^2)$
```

$\mathrm{J}/(\mathrm{mol}\,\mathrm{K})$
m/s^2

frac This leaves the `math.frac` as it is.

```
#metro-setup(frac-mode: "frac")
$unit(joule / (mole kelvin))$\
$unit(meter / second^2)$
```

$\frac{\mathrm{J}}{\mathrm{mol}\,\mathrm{K}}$
$\frac{\mathrm{m}}{\mathrm{s}^2}$

The following options affect only interpreted mode.

per-mode `Choice` (default: `"power"`)

Use to alter the handling of `per`.

power Reciprocal powers

```
#unit("#joule#per#mole#per#kelvin")\
#unit("#metre#per#second#squared")
```

$\mathrm{J\,mol^{-1}\,K^{-1}}$
$\mathrm{m\,s^{-2}}$

fraction Uses the `math.frac` function (also known as `$ / $`) to typeset positive and negative powers of a unit separately.

```
#unit("#joule#per#mole#per#kelvin", per-mode: "fraction")\
#unit("#metre#per#second#squared", per-mode: "fraction")
```

$\frac{\mathrm{J}}{\mathrm{mol\,K}}$
$\frac{\mathrm{m}}{\mathrm{s^2}}$

symbol Separates the two parts of a unit using the symbol in `per-symbol`. This method for displaying units can be ambiguous, and so brackets are added unless `bracket-unit-denominator` is set to `false`. Notice that `bracket-unit-denominator` only applies when `per-mode` is set to symbol.

```
#metro-setup(per-mode: "symbol")
#unit("#joule#per#mole#per#kelvin")\
#unit("#metre#per#second#squared")
```

$\mathrm{J/(mol\,K)}$
$\mathrm{m/s^2}$

bracket-unit-denominator `Switch` (default: `true`)

Whether or not to add brackets to unit denominators when `per-symbol` is `"symbol"`.

```
#unit("#joule#per#mole#per#kelvin", per-mode: "symbol", bracket-unit-denominator: false)
```

$\mathrm{J/mol\,K}$

sticky-per `Switch` (default: `false`)

Normally, `per` applies only to the next unit given. When `sticky-per` is `true`, this behaviour is changed so that `per` applies to all subsequent units.

```
#unit("#pascal#per#gray#henry")\
#unit("#pascal#per#gray#henry", sticky-per: true)
```

$\mathrm{Pa\,Gy^{-1}\,H}$
$\mathrm{Pa\,Gy^{-1}\,H^{-1}}$

qualifier-mode `Choice` (default: `"subscript"`)

Sets how unit qualifiers can be printed.

subscript

```
#unit("#kilogram#of(pol)#squared#per#mole#of(cat)#per#hour")
```

$\mathrm{kg_{pol}^2\,mol_{cat}^{-1}\,h^{-1}}$

bracket

```
#unit("#kilogram#of(pol)#squared#per#mole#of(cat)#per#hour", qualifier-mode: "bracket")
```

$\mathrm{kg(pol)^2\,mol(cat)^{-1}\,h^{-1}}$

combine Powers can lead to ambiguity and are automatically detected and brackets added as appropriate.

```
#unit("#deci#bel#of(i)", qualifier-mode: "combine")
```

dBi

/phrase: Used with `qualifier-phrase`, which allows for example a space or othre linking text to be inserted.

```
#metro-setup(qualifier-mode: "phrase", qualifier-phrase: sym.space)
#unit("#kilogram#of(pol)#squared#per#mole#of(cat)#per#hour")\
#metro-setup(qualifier-phrase: " of ")
#unit("#kilogram#of(pol)#squared#per#mole#of(cat)#per#hour")
```

kg pol^2 mol cat^{-1} h^{-1}
kg of pol^2 mol of cat^{-1} h^{-1}

power-half-as-sqrt `Switch` (default: `false`)

When `true` the power of 0.5 is shown by giving the unit sumbol as a square root.

```
#unit("#Hz#tothe(0.5)")\
#unit("#Hz#tothe(0.5)", power-half-as-sqrt: true)
```

$\mathrm{Hz}^{0.5}$
$\sqrt{\mathrm{Hz}}$

interpreted-delimeter `string` (default: #)

The delimeter to use to separate the macros in intepreted mode.

```
#unit("#joule#per#mole#per#kelvin")\
#unit("joule per mole per kelvin", interpreted-delimeter: " ")
```

J mol^{-1} K^{-1}
J mol^{-1} K^{-1}

Note that the first macro does not actually need the delimeter.

2.4 Quantities

3 Meet the Units

The following tables show the currently supported prefixes, units and their abbreviations. Note that unit abbreviations that have single letter commands are not available for import for use in literal mode as math mode accepts single letters.

Unit	Command	Symbol
ampere	`ampere`	A
candela	`candela`	cd
kelvin	`kelvin`	K
kilogram	`kilogram`	kg
metre	`metre`	m
mole	`mole`	mol
second	`second`	s

Table 1: SI base units.

Unit	Command	Symbol	Unit	Command	Symbol
becquerel	`becquerel`	Bq	newton	`newton`	N
degree Celsius	`degreeCelsius`	℃	ohm	`ohm`	Ω
coulomb	`coulomb`	C	pascal	`pascal`	Pa
farad	`farad`	F	radian	`radian`	rad
gray	`gray`	Gy	siemens	`siemens`	S
hertz	`hertz`	Hz	sievert	`sievert`	Sv
henry	`henry`	H	steradian	`steradian`	sr
joule	`joule`	J	tesla	`tesla`	T
lumen	`lumen`	lm	volt	`volt`	V
katal	`katal`	kat	watt	`watt`	W
lux	`lux`	lx	weber	`weber`	Wb

Table 2: Coherent derived units in the SI with special names and symbols.

Unit	Command	Symbol
astronomicalunit	`astronomicalunit`	au
bel	`bel`	B
dalton	`dalton`	Da
day	`day`	d
decibel	`decibel`	dB
degree	`degree`	°
electronvolt	`electronvolt`	eV
hectare	`hectare`	ha
hour	`hour`	h
litre	`litre`	L
	`liter`	L
minute (plane angle)	`arcminute`	′
minute (time)	`minute`	min
second (plane angle)	`arcsecond`	″
neper	`neper`	Np
tonne	`tonne`	t

Table 3: Non-SI units accepted for use with the International System of Units.

Prefix	Command	Symbol	Power	Prefix	Command	Symbol	Power
quecto	`quecto`	q	−30	deca	`deca`	d a	1
ronto	`ronto`	r	−27	hecto	`hecto`	h	2
yocto	`yocto`	y	−24	kilo	`kilo`	k	3
atto	`atto`	a	−18	mega	`mega`	M	6
zepto	`zepto`	z	−21	giga	`giga`	G	9
femto	`femto`	f	−15	tera	`tera`	T	12
pico	`pico`	p	−12	peta	`peta`	P	15
nano	`nano`	n	−9	exa	`exa`	E	18
micro	`micro`	µ	−6	zetta	`zetta`	Z	21
milli	`milli`	m	−3	yotta	`yotta`	Y	24
centi	`centi`	c	−2	ronna	`ronna`	R	27
deci	`deci`	d	−1	quetta	`quetta`	Q	30

Table 4: SI prefixes

Unit	Abbreviation	Symbol
femtogram	fg	fg
picogram	pg	pg
nanogram	ng	ng
microgram	ug	μg
milligram	mg	mg
gram	g	g
kilogram	kg	kg
picometre	pm	pm
nanometre	nm	nm
micrometre	um	μm
millimetre	mm	mm
centimetre	cm	cm
decimetre	dm	dm
metre	m	m
kilometre	km	km
attosecond	as	as
femtosecond	fs	fs
picosecond	ps	ps
nanosecond	ns	ns
microsecond	us	μs
millisecond	ms	ms
second	s	s
femtomole	fmol	fmol
picomole	pmol	pmol
nanomole	nmol	nmol
micromole	umol	μmol
millimole	mmol	mmol
mole	mol	mol
kilomole	kmol	kmol
picoampere	pA	pA
nanoampere	nA	nA
microampere	uA	μA
milliampere	mA	mA
ampere	A	A
kiloampere	kA	kA
microlitre	uL	μL
millilitre	mL	mL
litre	L	L
hectolitre	hL	hL
millihertz	mHz	mHz
hertz	Hz	Hz
kilohertz	kHz	kHz
megahertz	MHz	MHz
gigahertz	GHz	GHz
terahertz	THz	THz
millinewton	mN	mN
newton	N	N
kilonewton	kN	kN
meganewton	MN	MN
pascal	Pa	Pa
kilopascal	kPa	kPa
megapascal	MPa	MPa
gigapascal	GPa	GPa
milliohm	mohm	mΩ
kilohm	kohm	kΩ
megohm	Mohm	MΩ
picovolt	pV	pV
nanovolt	nV	nV
microvolt	uV	μV
millivolt	mV	mV
volt	V	V
kilovolt	kV	kV
watt	W	W
nanowatt	nW	nW
microwatt	uW	μW
milliwatt	mW	mW
kilowatt	kW	kW
megawatt	MW	MW
gigawatt	GW	GW
joule	J	J
microjoule	uJ	uJ
millijoule	mJ	mJ
kilojoule	kJ	kJ
electronvolt	eV	eV
millielectronvolt	meV	meV
kiloelectronvolt	keV	keV
megaelectronvolt	MeV	MeV
gigaelectronvolt	GeV	GeV
teraelectronvolt	TeV	TeV
kilowatt hour	kWh	kWh
farad	F	F
femtofarad	fF	fF
picofarad	pF	pF
nanofarad	nF	nF
microfarad	uF	μF
millifarad	mF	mF
henry	H	H
femtohenry	fH	fH
picohenry	pH	pH
nanohenry	nH	nH
millihenry	mH	mH
microhenry	uH	μH
coulomb	C	C
nanocoulomb	nC	nC
millicoulomb	mC	mC
microcoulomb	uC	μC
kelvin	K	K
decibel	dB	dB
astrnomicalunit	au	au
becquerel	Bq	Bq
candela	cd	cd
dalton	Da	Da
gray	Gy	Gy
hectare	ha	ha
katal	kat	kat
lumen	lm	lm
neper	Np	Np
radian	rad	rad
sievert	Sv	Sv
steradian	sr	sr
weber	Wb	Wb

Table 5: Unit abbreviations