

**SYNOPSIS**

rink [OPTIONS] [EXPR]...

**ARGS**

<EXPR>...      Evaluate a list of expressions. If no arguments are provided, an interactive session will start.

**OPTIONS**

          --config-path      Prints a path to the config file, then exits  
-f, --file <file>      Reads expressions from a file  
-h, --help              Print help information  
-V, --version            Print version information

**INTRODUCTION**

Rink is a unit conversion and calculation tool which can be used for both small and simple arithmetic and more complex dimensionality analysis and other tasks. In this manual, each section will have examples of the discussed material at the top.

**TABLE OF CONTENTS**

- 1. Working with units
  - 1. Weight vs Mass
- 2. Expressions
  - 1. Writing Numbers
  - 2. Multiplication
  - 3. Division
  - 4. Addition, Subtraction
  - 5. Temperatures
  - 6. Numbers
  - 7. Powers
  - 8. Inline unit definitions
  - 9. Custom base units
  - 10. Accessing the previous result
- 3. Units and prefixes
- 4. Top-level statements
  - 1. Conversions (->)
    - 1. Unit lists
    - 2. Temperature
    - 3. Bases
    - 4. Digits
  - 2. Unit finding
  - 3. Factorization
  - 4. Search
- 5. Dates
- 6. Substances
- 7. Index of Units
  - 1. Official SI Derived Units

2. Substances
3. Constants
4. Currencies
5. Functions

## WORKING WITH UNITS

In order to understand Rink, an understanding of units themselves is required. Fundamentally, a *unit* is a way of assigning a concrete value to a specific *quantity* such as length, volume, energy, power, current, etc. Each *quantity* can be reduced into other quantities (for example, area is length x length), except for 7 *base units*.

The 7 *base units* (as well as their SI *units*):

- Length (meters)
- Time (seconds)
- Mass (kilograms)
- Current (amperes)
- Temperature (kelvin)
- Amount of Substance (mol)
- Human-subjective brightness of light (candela)

In addition, Rink defines a few non-SI base units:

- Money (Euro)
- Information (bit)
- Radian
- Steradian
- Musical note length (wholenote)
- Biological activity (IU)

Each of these quantities is treated as irreducible. The 7 base units are the foundations of SI, and customary systems as well. (Customary systems are defined in terms of SI.)

Every *unit* is composed of two parts: A numerical value, and its *dimensionality*. The dimensionality is how a unit relates itself to the *base units*. Each base unit is raised to a certain power to construct the dimensionality. For example, the dimensionality of the quantity of acceleration is  $\text{length}^1 * \text{time}^{-2}$  and then the rest of the base units are to the 0th power, which is to say that they do not matter. Two units are considered *conformable* if they have matching dimensionalities, and they can then be used in conversions.

Because each unit has a numerical part, it is possible to do normal math on them.

- Adding two units produces a new unit with matching dimensionality.
- Multiplying two units produces a new unit with its dimensionality as each base unit multiplied together, e.g. velocity ( $\text{length time}^{-1}$ ) \* hertz ( $\text{time}^{-1}$ ) = acceleration ( $\text{length time}^{-2}$ ).
- Dividing two units is like multiplication, but taking away from the base units. A unit divided by itself is *dimensionless*, it has no quantity. Normal numbers are dimensionless.

Because of this, units are essentially just numbers. As such, Rink is essentially a calculator which takes dimensionality into account.

## WEIGHT VS MASS

It is important to remember the differences between mass and weight when working with mass and force units. Here are some tips:

- Mass doesn't change depending on the amount of gravity, and directly influences momentum.
- Weight is the amount of downward force on an object due to gravity.
- Mass is measured in kilograms or pounds.
- Weight is measured in newtons, kilogram force (kgf), or pound force (lbf).

- When someone says something weighs some amount of kg or lb, they're saying it has a weight of that number of kgf or lbf. This includes things like weight on the moon. (Don't correct anyone using this common figure of speech.)
- A scale displays an estimate of mass by measuring the force applied to it divided by its calibrated measurement of the acceleration of gravity. Its mass estimate would be incorrect on other planets unless it was recalibrated. You can also think of the displayed value as being weight in kgf or lbf.
- You can compute weight by multiplying mass by gravity. Both kgf and lbf have earth gravity as part of their definition, so when you multiply kg or lb by gravity you get the same numerical values back, but with kgf or lbf units.

EXPRESSIONS

WRITING NUMBERS

> 10.1e2  
1010 (dimensionless)  
> 10  
10 (dimensionless)  
> 0x10  
16 (dimensionless)  
> 0o10  
8 (dimensionless)  
> 0b10  
2 (dimensionless)

Decimal numbers can be written with an integer component, an after-decimal-point component, and an exponent. Numbers can optionally have either U+2009 THIN SPACE or an underscore ( \_ ) for digit place separators.

The decimal point is always written with a dot ( . ), not a comma or other marker. If the decimal point is provided, it must be followed by more digits. (1 . is not allowed.)

The exponent starts with an e, followed by an integer with an optional sign. The exponent is shorthand for writing out \* 10^exp. There can be no spaces within the number other than allowed digit separators. (10 e10 is not allowed.)

Hexadecimal, octal, and binary integers can be written using 0x, 0o, and 0b prefixes, respectively. These literals do not currently support decimal points or exponents.

MULTIPLICATION

> 3 4 m 5 s  
60 m s  
> 3 \* 4 m 5 s  
60 m s

Multiplication can be either by juxtaposition (that is, without any symbol) or using an explicit \* operator.

DIVISION

> 10 km / 5 m  
2000 (dimensionless)  
> 1|2 m  
0.5 m (length)

There are two division operators, for separate purposes. / has lower precedence than multiplication, and is used mainly for separating two halves of an entire expression. | has higher precedence than multiplication, and is used mainly for fractions of integers.

ADDITION, SUBTRACTION

> 1 \* 2 + 1 \* 2  
4 (dimensionless)  
> 12 meters + 5 feet  
13.524 m (length)

These have lower precedence than multiplication and division.

TEMPERATURES

> 12 °C  
285.15 K (temperature)

Temperature scales are operators with higher precedence than addition, and lower than multiplication. See the section on temperature conversions for more detailed syntax.

**NUMBERS**

```
> 9999999999999
approx. 9.99999e12 (dimensionless)
> 1.001
1.001 (dimensionless)
> 1e100
1.0e100 (dimensionless)
```

Numbers can be written with a fraction, and can be written in scientific notation. 1e24 is short for  $1 \times 10^{24}$ .

**POWERS**

```
> 12 ft^2
435483/390625, approx. 1.114836 m^2 (area)
```

Powers have higher precedence than multiplication. Both ^ and \*\* can be used.

**INLINE UNIT DEFINITIONS**

```
> ton -> rock = 10kg
45359237/500000, approx. 90.71847 rock (mass)
```

An equals expression is one which simultaneously defines a new unit with the right-hand side, names it using the left-hand side, and then produces it as its result. This is useful for customizing the output of the right-hand side of a conversion or converting into things that don't currently have units such as the amount of calories in a potato.

**CUSTOM BASE UNITS**

```
> 12 'core' hour / 3 'core' -> minutes
240 minutes (time)
```

A unit name which is wrapped in quotation marks will not be checked for whether it exists when it is evaluated. This means you can wrap anything in quotes to in effect produce a new *base unit* for the purposes of a single calculation. This can be useful for doing calculations in terms of things which are otherwise dimensionless.

**ACCESSING THE PREVIOUS RESULT**

```
> 100 ohm + 50 ohm
150 ohm (resistance)
> ANS * 10 mA
1.5 volt (electrical_potential)
```

The result of the previous query can be accessed with `_`, `ans` or `ANS`, which can be convenient for breaking up calculations into multiple steps. Note that when `rink` returns an error occurs, the previous result is kept. Also, currently only the results of mathematical expressions are stored, the results for conversions aren't.

**UNITS AND PREFIXES**

Units can be prefixed with SI prefixes as well as a number of non-SI prefixes, such as: quarter, double, kibi, mebi, %.

Rink will accept plural units.

**TOP-LEVEL STATEMENTS****CONVERSION (->)**

```
> meter -> feet
3.280839 foot (length)
> 12 °C -> °F
53.6 °F (temperature)
```

The left hand side of the conversion is an arbitrary expression, and the right hand side is one of:

- An arbitrary expression
- A temperature scale (celsius, fahrenheit, and several historical scales)
- A unit list (e.g. hour;min;sec)

**UNIT LISTS**

```
> 2^17 seconds -> hour;min;sec
36 hour, 24 minute, 32 s (time)
> 2 km -> mi;ft
1 mile, 1281.679 foot (length)
> liter -> cup;tbsp;tsp
```

4 uscup, 3 ustablespoon, 1.884136 usteaspoon (volume)

A unit list is a comma- or semicolon- delimited list of units with the same dimensionality, which can be used for breaking down numbers into more familiar quantities.

TEMPERATURE

> 12 °C  
285.15 K (temperature)  
> 12 degC  
285.15 K (temperature)  
> 12 celsius  
285.15 K (temperature)

Temperature scales in Rink are handled a little specially, because only Kelvin and Rankine (the absolute zero version of Fahrenheit) start at absolute zero. As such, they are *operators*, not units. These operators have looser binding precedence than multiplication, but tighter than addition.

Available temperature scales:

- degC, °C, celsius, : **Celsius**, the standard scale in most countries.
- degF, °F, fahrenheit, : **Fahrenheit**, the scale used in households across the United States.
- degRé, °Ré, degRe, °Re, réaumur, reaumur: **Réaumur**: A historical scale once used throughout Europe.
- degRø, °Rø, degRo, °Ro, rømer, romer: **Romer**: Yet another historical scale.
- degN, °N, degnewton: **Newton**: A historical scale created by Isaac Newton.
- degDe, °De, delisle: **Delisle**: A historical scale which, alongside the original Celsius scale, is reversed from the scales we are used to today. Its zero point is boiling water, and the freezing point of water is 150°De.

Note that these temperature scale measurements are *absolute* measurements, not *differences*. If you wish to say something like "a difference of 1°C", then you must use the absolute scale for the scale you're using. These are:

- For Celsius, kelvin K
- For Fahrenheit, Rankine degR
- For Réaumur, reaumur\_absolute (absolute as in the zero point is absolute zero)
- For Rømer, romer\_absolute
- For Newton, newton\_absolute
- For Delisle, delisle\_absolute

BASES

> 1000 -> hex  
3e8 (dimensionless)  
> 10000 -> base 36  
7ps (dimensionless)  
> pi meter -> hex meter  
approx. 3.243f6a meter (length)

Base modifiers are specified with base followed by a number, followed by the rest of your conversion. Allowed bases are currently 2 through 36. There are some special base names which are also recognized:

- hex, hexadecimal, base16 for base 16.
- oct, octal, base8 for base 8.
- bin, binary, base2 for base 2.

DIGITS

> 2^128 -> digits  
340282366920938463463374607431768211456 (dimensionless)  
> 1/7 -> digits 50  
1/7, approx. 0.1428571428571428571428571428571428571428571428 (dimensionless)  
> googol -> digits

$$()$$

Digits modifiers are specified with `digits` optionally followed by a number, before the base modifier and before the rest of the conversion. Any number of digits are allowed, but large amounts may not succeed.

The function of this modifier is that it forces the entire integer part to be printed (i.e., scientific notation will *not* be used), and then it prints an additional  $n$  digits, using the default if not specified.

Trancendental numbers currently cannot be precisely represented, so asking for many digits of pi or e will produce unsatisfying results.

Trigonometric and logarithmic functions are currently implemented using a machine-float fallback, because their results cannot be precisely represented as finite rationals. Because of this, asking for many digits of such numbers will also produce unsatisfying results.

```
> units for power
Units for kg m^2 / s^3 (power): VA, airwatt, boilerhorsepower, brhorsepower,
donkeypower, electrichorsepower, horsepower, lusec, mbh, metrichorsepower,
poncelet, sccm, sccs, scfh, scfm, slph, slpm, solarluminosity,
tonrefrigeration, waterhorsepower, watt
```

The `units` `for`, `units` `of`, and `units` commands will find more units which match the dimensionality of the one specified.

```
> factorize velocity
Factorizations: velocity; frequency length; area viscosity;
acceleration time; length^2 viscosity
> factorize power
Factorizations: power; force velocity; radiant_intensity solid_angle;
area radiosity; length spectral_flux_wavelength; radiation_dose spectral_exposure_frequency;
spectral_irradiance_wavelength volume; temperature thermal_conductance;
energy frequency; current^2 resistance; ...
```

Unit factorization is what Rink names the process of finding quantities which can be multiplied together to produce the original quantity. This can be useful for discovering new ways to construct a unit.

```
> search milk
Search results: milk (density), mil (length), mile (length), mill (dimensionless), mi (length)
```

Allows you to search for units based on their names, returning up to 5 results and showing the associated physical quantity of the unit.

```
> #jan 01, 1970#
1970-01-01 00:00:00 +00:00 (46 years ago)
> now - #jan 01, 1970# -> gigaseconds
1.472083 gigaseconds (time)
> #2016-08-24# + 500 weeks
2026-03-25 00:00:00 +00:00 (in 9 years)
```

In addition to handling units, Rink is also capable of doing some calculations with dates and times.

### Converting to a timezone:

```
> now
2022-08-08 21:19:56.990897100 -07:00 (now)
> now -> "Europe/London"
2022-08-09 05:20:03.656075600 BST (now)
```

Converting to a fixed offset:

```
> now -> +01:00
2022-08-09 05:20:30.080703900 +01:00 (now)
```

Inputting a time with an offset:

```
> #apr 1, 2016 12:00:00 +01:00#
2016-04-01 12:00:00 +01:00 (6 years ago)
```

## SUBSTANCES

```
> milk
milk: density = 242 gram -> approx. 236588.2 millimeter^3
> gallon milk
milk: volume = approx. 3785411.7 millimeter^3; mass = 3.872 kilogram
> egg
egg: USA large egg. mass_shelled = 50 gram; mass_white = 30 gram;
mass_yolk = 18.6 gram; volume = approx. 46824.75 millimeter^3;
volume_white = approx. 29573.52 millimeter^3;
volume_yolk = approx. 17251.22 millimeter^3
> egg_shelled of kg egg
20 (dimensionless)
> gallon gasoline -> btu
gasoline: volume = approx. 3785411.7 millimeter^3; energy_HHV = 125000 btu; energy_LHV = 115000 btu
```

Substances are how Rink organizes the physical properties of materials, objects, both countable and uncountable. Each substance has a name, an associated amount (defaulting to dimensionless 1), and a set of associated properties.

Each property maps a named input into a named output and vice versa, and has a name itself. Countable objects often have properties with an input being dimensionless, so that you do not need to specify an amount to extract the property.

The properties of a substance are accessed with the `of` operator (*property of substance*), which reads a multiplication expression following it, so you may have to wrap it in parentheses.

Substances can be used in conversions, and can be added and multiplied to transform them. Multiplication will change the amount of the substance you have, so that you can write `kg egg` to specify one kilogram of eggs. Addition will combine certain properties (currently only `molar_mass`) to create a new substance entirely. Conversions of substances allow you to get multiple results simultaneously, for example if there are multiple different measurements of some property of the substance available.

If the result of a calculation results in a substance, Rink will show all of the properties applicable for the given amount.

## INDEX OF UNITS

The full list of units is specified in the file `definitions.units` (<https://github.com/tiffany352/rink-rs/blob/master/definitions.units>), but a small list of the most helpful ones will be listed here. It is intended that most units should be easy to guess the names of.

### OFFICIAL SI DERIVED UNITS

- Newton N (force)
- Pascal Pa (pressure)
- Joule J (energy)
- Watt W (power)
- Coulomb C (charge)
- Volt V (electrical potential)
- Ohm (electrical resistance)
- Siemens S (electrical conductance)
- Farad F (capacitance)
- Weber Wb (magnetic flux)
- Henry H (inductance)
- Tesla T (magnetic flux density)
- Hertz Hz (frequency)
- Lumen lm (luminous flux)
- Lux lx (illuminance)

- Gray Gy (radiation dose)
- Katal kat (catalytic activity)

**SUBSTANCES**

- Water water
- Mercury (Hg) mercury
- Compounds and materials
  - Ammonia ammonia
  - Freon freon
  - Tissue tissue
  - Diamond diamond
  - Graphite graphite
  - Water ice ice
  - Asphalt asphalt
  - Brick brick
  - Concrete cocnrete
  - Silica glass glass\_silica
  - Flint glass glass\_flint
  - Pyrex glass glass\_pyrex
  - Gypsum gypsum
  - Marble marble
  - Sand sand
  - Soil soil
  - Air air
- Particles
  - Electron electron
  - Proton proton
  - Neutron neutron
  - Deuterium nucleus deuteron
  - Muon muon
  - Helium nucleus helion
  - Tau tauparticle
  - Alpha alphaparticle
  - Tritium nucleus triton
- Celestial bodies
  - Sun sun
  - Mercury mercury\_planet
  - Venus venus
  - Earth earth
  - Earth's moon moon



- Mars mars
- Jupiter jupiter
- Saturn saturn
- Uranus uranus
- Neptune neptune
- Pluto pluto
- Fuels
  - Crude oil oil
  - Coal coal
  - Natural gas naturalgas
  - Charcoal charcoal
  - Wood wood
  - Ethanol ethanol
  - Diesel diesel
  - Gasoline gasoline
  - Heating oil heating\_oil
  - Fuel oil fueloil
  - Propane propane
  - Butane butane
- Foods
  - Butter butter
  - Clarified butter butter\_clarified
  - Cocoa butter cocoa\_butter
  - Vegetable shortening shortening
  - Vegetable oil vegetable\_oil
  - Olive oil olive\_oil
  - Flour cakeflour, flour, breadflour
  - Corn starch cornstarch
  - Cocoa dutchcocoa, cocoa
  - Heavy cream heavycream
  - Milk milk
  - Sour cream sourcream
  - Molasses molasses
  - Corn syrup corrn syrup
  - Honey honey
  - Sugar sugar
  - Powdered sugar powdered\_sugar
  - Brown sugar brownsugar\_light, brownsugar\_dark

- Baking powder `baking_powder`
- Salt `salt`, `koshersalt`
- Egg `egg`
- Elements 1 through 118, by name (e.g. `helium`)

**CONSTANTS**

- Pi `pi`
- Speed of light `c`
- Planck Constant `planck_constant`
- Gravitational Constant `G`
- Avogadro's number `avogadro`
- Gas Constant `gasconstant`
- Boltzmann Constant `boltzmann`
- Earth Gravity `gravity`, `force`
- Earth Atmosphere Density `atm`

**CURRENCIES**

These are only enabled if Rink was compiled with support for currencies.

- EU Euro `EUR`, `€`
- US dollar `USD`, `$`, `dollar`
- Japan yen `JPY`, `¥`, `yen`
- Bulgaria lev `BGN`
- Czech koruna `CZK`
- Denmark kroner `DKK`
- UK pound `GBP`, `£`
- Hungary forint `HUF`
- Poland złoty `PLN`
- Romania lei `RON`
- Sweden krona `SEK`
- Switzerland franc `CHF`
- Norway krone `NOK`
- Croatia kuna `HRK`
- Russia ruble `RUB`,
- Turkey lira `TRY`,
- Australia dollar `AUD`, `A$`
- Brazil real `BRL`, `R$`
- Canada dollar `CAD`, `C$`
- PRC yuan `CNY`
- Hong Kong dollar `HKD`, `H$`
- Indonesia rupiah `IDR`
- Israel shekel `ILS`,

- India rupee INR,
- South Korea won
- Mexico dollar MXN, mex\$
- Malaysia ringgit MYR
- New Zealand dollar NZD, NZ\$
- Phillipines piso PHP,
- Singapore dollar SGD, S\$
- Thailand baht THB,
- South Africa rand ZAR

## FUNCTIONS

Currently, all of these result in machine float fallback, because their results are real numbers that cannot be precisely represented as rationals.

- `sqrt (x)` : Square root,  $\sqrt{x}$ .
- `exp (x)` : The exponential function,  $e^x$ .
- `ln (x)` : The natural logarithm,  $\log_e(x)$ .
- `log (x, y)` : Logarithm in base y,  $\log_y(x)$ .
- `log2 (x)` : Logarithm in base 2,  $\log_2(x)$ .
- `log10 (x)` : Logarithm in base 10,  $\log_{10}(x)$ .
- `hypot (x, y)` : The length of the hypotenuse of a right-angle triangle given adjacent edges of length x and y.
- `sin (x)` : The sine function.
- `cos (x)` : The cosine function.
- `tan (x)` : The tangent function.
- `asin (x)` : Inverse sine, or arcsine.
- `acos (x)` : Inverse cosine, or arccosine.
- `atan (x)` : Inverse tangent, or arctangent.
- `atan2 (x, y)` : Four-quadrant arctangent, which can be used to reverse sine+cosine back into an angle.
- `sinh (x)` : Hyperbolic sine.
- `cosh (x)` : Hyperbolic cosine.
- `tanh (x)` : Hyperbolic tangent.
- `asinh (x)` : Inverse hyperbolic sine function.
- `acosh (x)` : Inverse hyperbolic cosine function.
- `atanh (x)` : Inverse hyperbolic tangent function.