

1. Basic Conversions

This package defines units, e. g. #kilogram or #watt. It is possible to define custom shorthands for units as well as custom quantities:

```
let kwh = multiply(kilo, watt, hour)
let mj = multiply(mega, joule)
let myenergyqty = multiply(15, kwh)
```

As you saw here, each unit already has an implicit value of 1.

Using the `value-in(qty, unit)` function, a quantity is converted to another unit, provided that the physical dimensions are equal:

54

Furthermore, units may be prepared for formatting using the `fmt(qty, unit, digits)` function. The resulting strings can then be passed to Christian Hecker's unify package [1] which provides the formatting function (`qty`):

```
import "@preview/unify:0.7.1": qty
$qty(..fmt(myenergyqty)) = qty(..fmt(myenergyqty, unit: mj))$
```

15 kW h = 54 MJ

If no unit for formatting is specified, we use the unit, that was originally used for defining the quantity.

The result may be automatically rounded to a specified number of digits:

```
let ms2 = divide(meter, pow(second, 2))
let mygravity = multiply(9.80665, ms2)
$qty(..fmt(mygravity)) approx #qty(..fmt(mygravity, digits: 1))$
//                               ^^ make sure to add
```

9.806 65 m s⁻² ≈ 9.8 m s⁻²

Note that `typst` parses numbers inside math mode as content. However, we cannot use content as an argument to any of the functions. To correct for the wrong parsing, either

- begin one of the function calls with #
- escape the number with #

We can also perform maths using the `add`, `subtract`, `multiply`, `divide`, `pow` functions:

```
let myvolt = multiply(115, volt)
let myohm = multiply(220, kilo, ohm)
let mycurrent = divide(myvolt, myohm)
```

When applying

```
#qty(..fmt(myvolt))
on a
#qty(..fmt(myohm, unit: multiply(mega, ohm)))
resistor, the current is
#qty(..fmt(mycurrent, unit: multiply(milli, ampere), digits: 3))
```

When applying 115 V on a 0.22 MΩ resistor, the current is 0.523 mA.

2. A Complete Example: Switching Converter

2.1. Context

When calculating switching converters, there are many units and even more prefixes. In this example, we will calculate an exemplary switching calculator:

- Input voltage is given as $V_I = 5 \text{ V}$
- Output voltage is given as $V_O = 2\,200 \text{ mV} = 2.2 \text{ V}$
- Switching frequency is determined by the controller which uses $f_S = 300 \text{ kHz}$
- Output current is given as $I = 10 \text{ A}$ and inductor current ripple ratio shall be $CR = 30\%$ [2]
- The output voltage ripple shall be less than $\Delta V_O = 10 \text{ mV}$.

2.2. Basics

First, we define all our known values. From there, we can already perform some basic calculations:

```
let vi = multiply(5, volt)
let vo = multiply(2200, milli, volt)
let fs = multiply(300, kilo, hertz)
let io = multiply(10, ampere)
let cr = multiply(30, percent)
```

```
let dcycle = divide(vo, vi)
let ir = multiply(io, cr)
```

The duty cycle can be calculated to $D = V_O/V_I = \text{qty}(\text{..fmt}(\text{dcycle}, \text{unit: \#1}))\$$.

The current ripple would be $\Delta I_L = \text{qty}(\text{..fmt}(\text{ir}, \text{unit: ampere}))\$$.

The duty cycle can be calculated to $D = \frac{V_O}{V_I} = 0.44$.

The current ripple would be $\Delta I_L = 3 \text{ A}$.

2.3. Complex Calculations

Calculation of the inductance and output capacitor is a more complex calculation [3]:

$$L = \frac{D(V_I - V_O)}{\Delta I_L f_S} \quad C_O = \frac{\Delta I_L}{8 f_S \Delta V_O}$$

```
let inductance = divide(multiply(dcycle, subtract(vi, vo)), multiply(ir, fs))
let capacitance = divide(ir, multiply(8, fs, dvo))
```

With a simple calculator, one would get results such as

$$L = 0.013\,688\,888\,888\,888\,89 \text{ mV V}^{-1} \text{ V A}^{-1} \%^{-1} \text{ kHz}^{-1}$$
$$C_O = 0.012\,5 \text{ A \% kHz}^{-1} \text{ mV}^{-1}$$

Which this library can simply convert using:

```
$ L &= qty(..fmt(inductance, unit: multiply(micro, henry), digits: #3)) \
C_0 &= qty(..fmt(capacitance, unit: multiply(micro, farad), digits: #3)) $
```

$$L = 1.369 \mu\text{H}$$
$$C_O = 125 \mu\text{F}$$

3. Example: Stress Analysis

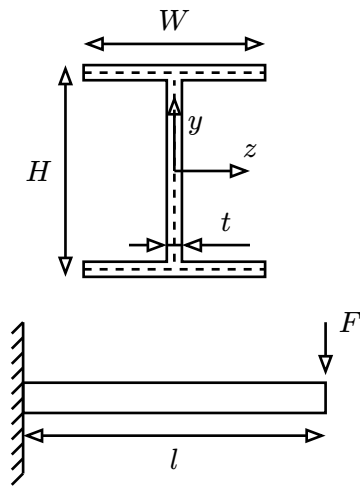


Figure 1: I-beam profile (simplified)

A simple I beam is defined using

- Height $H = 140$ mm
- Width $W = 120$ mm
- Constant thickness $t = 8$ mm $\ll H, W$
- Beam length $l = 3$ m
- Applied force at the end $F = 16$ kN
- Yield strength $\sigma_y = 355$ MPa (S355 material)

The area moment of inertia may be approximated in thin-walled profiles like this:

$$\begin{aligned} I_{yy} &= \int x^2 da = t \int x^2 ds \\ &= 2t \int_{-\frac{W}{2}}^{+\frac{W}{2}} s^2 ds = 2t \left[\frac{1}{3} s^3 \right]_{-\frac{W}{2}}^{+\frac{W}{2}} = 2t \cdot \frac{1}{3} \cdot 2 \cdot \frac{W^3}{8} = \underline{\underline{\frac{W^3 t}{6}}} \end{aligned}$$

As well as

$$I_{zz} = t \int y^2 ds = t \left(\frac{WH^2}{2} + \frac{1}{3} \cdot 2 \cdot \frac{W^3}{8} \right) = \underline{\underline{\frac{WH^2 t}{2} + \frac{W^3 t}{12}}}$$

Using the specific sizes, we calculate $I_{yy} = 2\,304\,000 \text{ mm}^4$ and $I_{zz} = 10\,560\,000 \text{ mm}^4$.

The section modulus measures the resistance against a bending moment: $W_y = \frac{I_{yy}}{W/2} = 38\,400 \text{ mm}^3$ and $W_z = \frac{I_{zz}}{H/2} = 150\,857 \text{ mm}^3$.

Now we can calculate the stress under load: $\sigma = \frac{M}{W} = \frac{F \cdot l}{W} = 318 \text{ MPa}$.

In this scenario, the reserve factor were $RF = \frac{\text{strength}}{\text{load}} = 111.6 \%$.

4. Defining Custom Units

4.1. Defining Imperial Units

We can define custom units based on existing units using `define_unit`:

```
let inch = define_unit(multiply(25.4, milli, meter), "in ")
let foot = define_unit(multiply(12, inch), "ft ")
let yard = define_unit(multiply(3, foot), "yd ")
let mile = define_unit(multiply(1760, yard), "mi ")
```

Furthermore, when introducing new units, make sure that they either exist in `unify`, or that you register them there as well [1]:

```
import "@preview/unify:0.7.1": add-unit
add-unit("inch", "in", "upright(\"in\")")
add-unit("feet", "ft", "upright(\"ft\")")
add-unit("yard", "yd", "upright(\"yd\")")
add-unit("mile", "mi", "upright(\"mi\")")
```

Afterwards, it is no more problem to convert these units:

```
let mydist = multiply(3, mile)
$qty(..fmt(mydist, unit: mile)) =
qty(..fmt(mydist, unit: foot)) =
qty(..fmt(mydist, unit: meter, digits: #2)) =
qty(..fmt(mydist, unit: multiply(kilo, meter), digits: #3))$
```

3 mi = 15 840 ft = 4 828.03 m = 4.828 km

4.2. Using Custom Units in Powers

This also works with squared and cubed units:

```
define_unit(pow(multiply(deci, meter), 3), "L ")
#let myvolume = multiply(10, pow(yard, 3))
```

A concrete truck carries `$qty(..fmt(myvolume)) =`
`qty(..fmt(myvolume, unit: pow(meter, #3), digits: #2)) =`
`qty(..fmt(myvolume, unit: liter, digits: #0))$` of concrete.

A concrete truck carries $10 \text{ yd}^3 = 7.65 \text{ m}^3 = 7\,646 \text{ L}$ of concrete.

4.3. Common Mistake

The second argument is the unit format, which is at some point passed to `unify`. When defining a unit instead of a prefix, you must add a space after the formatting string.

The library has no way of verifying that the unit format is correct, it is possible to do something like:

```
let inch = define_unit(multiply(25.4, second), "in ")
//                ^^^^^^  ^^^^^ undetectable mismatch
```

And the library has no way of detecting that error!

4.4. Pressure Conversion

First, we define some alternative units for pressure measurement:

```
let bar = define_unit(multiply(100, kilo, pascal), "bar ")
let atm = define_unit(multiply(1.01325, bar), "atm ")
let hecto = define_unit(100, "h") // for hPa
```

We can use these units to directly convert the standard atmospheric pressure [4]:

```
let pres = multiply(1, atm)
$qty(..fmt(pres)) =
qty(..fmt(pres, unit: bar)) =
qty(..fmt(pres, unit: multiply(hecto, pascal)))$
```

1 atm = 1.013 25 bar = 1 013.25 hPa

Furthermore, using the density of mercury, we can convert the pressure to the height of the liquid column:

```
let density = divide(multiply(13595.1, kilogram), pow(meter, 3))
let gravity = multiply(9.81, divide(meter, pow(second, 2)))
let hg = multiply(density, gravity)
let height = divide(pres, hg)
```

`qty(..fmt(pres))` is equivalent to
`qty(..fmt(height, unit: multiply(milli, meter), digits: 0))` or
`qty(..fmt(height, unit: inch, digits: 2))` of mercury.

1 atm is equivalent to 760 mm or 29.91 in of mercury.

Bibliography

- [1] Christian Hecker, *Format numbers, units, and ranges in Typst correctly*. [Online]. Available: <https://github.com/ChHecker/unify>
- [2] Frederik Dostal, "Selecting the Right Inductor Current Ripple," Mar. 06, 2023. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/selecting-the-right-inductor-current-ripple.html>
- [3] Brigitte Hauke, "Basic Calculation of a Buck Converter's Power Stage," Dec. 2011. [Online]. Available: <https://www.ti.com/lit/an/slva477b/slva477b.pdf>
- [4] "ISO 2533:1975 -- Standard Atmosphere." 1975.