The LUA-PHYSICAL library

Version 1.0.3

Thomas Jenni

September 9, 2020

Abstract

lua-physical is a pure Lua library, which provides functions and objects for the computation of physical quantities. A physical quantity is the product of a numerical value and a physical unit. The package has been written, to simplify the creation physics problem sets. The package provides units of the SI and the imperial system. Furthermore, an almost complete set of international currencies are supported, however without realtime exchange rates. In order to display the numbers with measurement uncertainties, the package is able to perform gaussian error propagation.

Contents

1 Introduction 3
  1.1 Dependencies .................................................. 3

2 Loading 3
  2.1 License ......................................................... 4

3 Usage 5
  3.1 Unit conversion .................................................. 5
    3.1.1 Temperature Conversion .................................... 6
  3.2 Uncertainty Propagation ....................................... 7
  3.3 Mathematical operations ....................................... 10

4 Supported Units 12
  4.1 Prefixes .......................................................... 12
  4.2 Base Units ....................................................... 13
  4.3 Constants ........................................................ 15
  4.4 Coherent derived units in the SI .............................. 16
  4.5 Non-SI units accepted for use with the SI .................... 17
  4.6 Nominal Astronomical Units .................................... 17
  4.7 Other Non-SI units .............................................. 18
1 Introduction

The author of this package is a physics teacher at the high school Kantonsschule Zug, Switzerland. The main use of this package is to write physics problem sets. Lua\LaTeX{} does make it possible to integrate physical calculations directly. The package has been in use since 2016. Many bugs have been found and fixed. Nevertheless it still is possible, that some were not found yet. Therefore the author recommends not to use this package in industry or science. If one does so, it’s the responsibility of the user to check results for plausibility. If the user finds some bugs, they can be reported at github.com.

1.1 Dependencies

In principle this library is standalone, but it is compatible with the siunitx package. Calculation results can be written to Lua\LaTeX{} directly by calling the tosiunitx() method. If the preamble in the next section is used, the printing is done by the \texttt{\textbackslash q}{} macro.

2 Loading

This package is a pure Lua library. Therefore one has to require it explicitly by calling \texttt{require("physical")}. For printing physical quantities, the siunitx is supported. It’s recommended to define a macro like \texttt{\textbackslash q} to convert the Lua quantity object to a siunitx expression.

The following \LaTeX{} preamble loads the lua-physical package and creates a macro \texttt{\textbackslash q} for printing physical quantities.

Listing 1: basic preamble

\begin{verbatim}
\usepackage{luacode}
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {.},
  per-mode = symbol,
  separate-uncertainty = true,
  add-decimal-zero = true,
  exponent-product = \cdot,
  round-mode = off
}
\begin{luacode*}
physical = require("physical")
N = physical.Number
\end{luacode*}
\end{verbatim}
2.1 License

This code is freely distributable under the terms of the MIT license.

Copyright (c) 2020 Thomas Jenni

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
3 Usage

Given the basic preamble, units can be used in lua code directly. By convention, all units have an underscore in front of them, i.e. meter is _m, second is _s. All available units are listed in section 4. The following example illustrates the use of this library.

Listing 2: Velocity of a car.

\begin{luacode}
s = 10 * _m\nt = 2 * _s\nv = s/t\n\end{luacode}

A car travels $s$ in $t$. calculate its velocity.

\begin{equation*}
v = \frac{s}{t} = \frac{10\text{ m}}{2\text{ s}} = 5.0\text{ m/s}
\end{equation*}

In the above listing the variable $s$ stands for displacement and has the unit meter _m, the variable $t$ stands for time and is given in seconds _s. If mathematical operations are done on them, new physical quantities are created. In the problem above, the velocity $v$ is calculated by dividing $s$ by $t$. The instance $v$ has the derived unit m/s. By using the macro \q{} all quantities can be printed to the LuaLATEX code directly.

3.1 Unit conversion

Very often, the result of a calculation has to be converted to another unit. In listing the task is to calculate the volume of a cuboid. The length of the edges are given in different units. The volume is calculated by multiplying all three lengths, the unit of the result is cm mm m. If the unit cm$^3$ is preferred, it has to be converted explicitly. The conversion function is called to() and is available on all physical quantity instances. At first this looks a bit cumbersome. The reason of this behaviour is, that the software is not able to guess the unit of the result. In many cases, like in the example here, it’s not clear what unit the result should have. Therefore the user has always to give the target unit explicitly.
Listing 3: Volume of a cuboid.

\begin{luacode}
a = 12 * _cm
b = 150 * _mm
c = 1.5 * _m

V = a*b*c
\end{luacode}

Find the volume of a rectangular cuboid with lengths $a$, $b$ and $c$.
\begin{equation*}
V = a \cdot b \cdot c = \q{a} \cdot \q{b} \cdot \q{c} = \underline{\q{V}:to(_{dm^3})}
\end{equation*}

Find the volume of a rectangular cuboid with lengths 12 cm, 150 mm and 1.5 m.

\[
V = a \cdot b \cdot c = 12 \text{ cm} \cdot 150 \text{ mm} \cdot 1.5 \text{ m} = 2700.0 \text{ cm mm m} = 27.0 \text{ dm}^3
\]

3.1.1 Temperature Conversion

Most physical units transform linearly. Exceptions are the unit degree Celsius \_degC and degree Fahrenheit \_degF. These units are ambiguous and can be interpreted as temperature differences or as an absolute temperatures. In the latter case, the conversion to base units is not a linear, but an affine transformation. This is because degree Celsius and degree Fahrenheit scales have their zero points at different temperatures compared to the unit Kelvin.

By default \_degC and \_degF units are temperature differences. If one wants to have it converted absolutely, it has to be done adding / subtracting \_degC\_0 = 273.15*\_K or \_degF\_0 = (273.15 - 32*(5/9)) * \_K, the zero point temperatures of the scales.

In the following problem, listing 4, the task is to convert temperatures given in the unit degree Celsius and degree Fahrenheit to Kelvin.

Listing 4: Temperature conversion.

\begin{luacode}
theta_1 = 110 * _degC
T_1 = ( theta_1 + _degC\_0 ):to(_K)
T_2 = 100 * _K
theta_2 = ( T_2 - _degC\_0 ):to(_degC)
\end{luacode}
\begin{align*}
\theta_3 &= 212 \cdot \text{degF} \\
T_3 &= (\theta_3 \cdot \text{degF}_0) : (\_K) \\
T_4 &= 100 \cdot \_K \\
\theta_4 &= (T_4 \cdot \text{degF}_0) : (\_degF) \\
\theta_5 &= 100 \cdot \text{degC} \\
\theta_6 &= ((\theta_5 \cdot \text{degC}_0) : (\_K) - \text{degF}_0) : (\_degF) \\
\end{align*}

\[110°C \equiv 383.15 K\]
\[-173.15°C \equiv 100 K\]
\[212°F \equiv 373.15 K\]
\[-279.67°F \equiv 100 K\]
\[100°C \equiv 212.0°F\]

3.2 Uncertainty Propagation

The package supports uncertainty propagation. To create a number with an uncertainty, an instance of \texttt{physical.Number} has to be created, see listing 5. It has to be remembered, that \texttt{N} is a alias for \texttt{physical.Number}. The first argument of the constructor \texttt{N(mean, uncertainty)} is the mean value and the second one the uncertainty of the measurement. If the proposed preamble \[2] is used, the uncertainty is by default separated from the mean value by a plus-minus sign.

For the uncertainty propagation the gaussian formula

\[\Delta f = \sqrt{\left(\frac{\partial f}{x_1} \cdot \Delta x_1\right)^2 + \cdots + \left(\frac{\partial f}{x_n} \cdot \Delta x_2\right)^2}\]

is used. This formula is a good estimation for the uncertainty \(\Delta f\), if the quantities \(x_1, \ldots, x_n\) the function \(f\) depends on, have no correlation. Further, the function \(f\) has to change linear, if quantities \(x_i\) are changed in the range of their uncertainties.
Listing 5: Uncertainty in area calculation.

```latex
\begin{luacode}
a = N(2,0.1) * _m
b = N(3,0.1) * _m
A = (a*b):to(_m^2)
\end{luacode}

Calculate the area of a rectangle with lengths $q{a}$ and $q{b}$.

\begin{equation*}
A = a \cdot b = q{a} \cdot q{b} = \underline{q{A}}
\end{equation*}

Instead of printing the uncertainties, one can use the uncertainty calculation to provide significant digits and omit it.

In the following problem, listing 6, the task is to find the volume of an ideal gas. Given are pressure $p$ in _bar_, amount of substance $n$ in _mol_ and temperature $T$ in degree celsius _degC_. In order to do the calculation, one has to convert $T$, which is given as an absolute temperature in degree celsius to the base unit Kelvin first. By setting $N\text{.omitUncertainty} = true$, all uncertainties are not printed.

Listing 6: Volume of an ideal gas.

```latex
\begin{luacode}
N\text{.omitUncertainty} = true
p = N(1.013,0.0001) * _bar
n = N(1,0.01) * _mol
T = N(30,0.1) * _degC
V = ( n \cdot _R \cdot (T + _degC_0):to(_K) / p ):to(_L)
\end{luacode}

An ideal gas ($q{n}$) has a pressure $q{p}$ and a temperature of $q{T}$. Calculate the volume of the gas.

\begin{equation*}
V = \frac{q{n} \cdot _R \cdot q{T}}{q{p}} = \underline{q{V}}
\end{equation*}
An ideal gas (1.0 mol) has a pressure of 1.013 bar and a temperature of 30°C. Calculate the volume of the gas.

\[ V = \frac{1.0 \text{ mol} \cdot 8.31 \text{ J/(mol K)} \cdot 303 \text{ K}}{1.013 \text{ bar}} = 25 \text{ L} \]

This example shows, that the result has only two digits. If more digits are needed, the uncertainties of the given quantities should be smaller.
3.3 Mathematical operations

Two physical quantities with identical dimensions can be added or subtracted. The library checks the validity of those operations and throws an error if two addends haven’t the same dimensions.

Listing 7: Addition and Subtraction

```
l_1 = 1 * _m
l_2 = 2 * _cm
t = 2 * _s

l_1 + t
Error: Cannot add '1* _m' to '2 * _s', because they have different dimensions.

l_1 + l_2
102.0 * _cm
```

New physical quantities can be created by division and multiplication. As long as no division by zero is made, no errors should occur.

Listing 8: Multiplication and Division

```
l_1 = 1 * _m
l_2 = 2 * _cm

(l_1 * l_2):to(_m^2)
0.02 * _m^2

(l_1 / l_2):to(_1)
50.0 * _1
```

Physical quantities can be exponentiated. The library doesn’t check, if the result has units with non integer exponents.

Listing 9: Exponentiation

```
l = 5 * _m
A = l^2

A:to(_m^2)
25.0 * _m^2

A:sqrt()
5.0 * _m

A^0.5
5.0 * _m
```
Exponential functions and the logarithms should have dimensionless arguments. The library throws an error if that’s not the case.

Listing 10: Exponential function and logarithm

```python
N_0 = 1000 * _1
lambda = Q.log(2)/(2*_h)
t = 50 * _min

N_0 * Q.exp(-lambda * t)
749.15353843834 * _1
```
4 Supported Units

All supported units are listed in this section. Subsection 4.2 lists the seven base units of the International System of Units (SI). In subsection 4.3, mathematical and physical constants are defined. The subsection 4.4 contains all coherent derived units from the SI system and 4.5 those which are accepted to use with the SI. The subsection 4.6 lists nominal astronomical units, which are proposed by [4]. Subsection 4.7 lists units, which are common but outside of the SI system. The subsections 4.8 and 4.9 are dedicated to imperial and U.S. customary units. The last subsection 4.10 contains international currencies.

4.1 Prefixes

All SI units have prefixed versions, i.e. _us microsecond, _cm centimeter, _mN millinewton, see table 1. Some units of data processing, like _bit have prefixes which are powers of 2. They are called binary or IEC prefixes, see table 2 [2, 121].

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>yotta</td>
<td>Y</td>
<td>1e24</td>
</tr>
<tr>
<td>zetta</td>
<td>Z</td>
<td>1e21</td>
</tr>
<tr>
<td>exa</td>
<td>E</td>
<td>1e18</td>
</tr>
<tr>
<td>peta</td>
<td>P</td>
<td>1e15</td>
</tr>
<tr>
<td>tera</td>
<td>T</td>
<td>1e12</td>
</tr>
<tr>
<td>giga</td>
<td>G</td>
<td>1e9</td>
</tr>
<tr>
<td>mega</td>
<td>M</td>
<td>1e6</td>
</tr>
<tr>
<td>kilo</td>
<td>k</td>
<td>1e3</td>
</tr>
<tr>
<td>hecto</td>
<td>h</td>
<td>1e2</td>
</tr>
<tr>
<td>deca</td>
<td>da</td>
<td>1e1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>deci</td>
<td>d</td>
<td>1e-1</td>
</tr>
<tr>
<td>centi</td>
<td>c</td>
<td>1e-2</td>
</tr>
<tr>
<td>milli</td>
<td>m</td>
<td>1e-3</td>
</tr>
<tr>
<td>micro</td>
<td>u</td>
<td>1e-6</td>
</tr>
<tr>
<td>nano</td>
<td>n</td>
<td>1e-9</td>
</tr>
<tr>
<td>pico</td>
<td>p</td>
<td>1e-12</td>
</tr>
<tr>
<td>femto</td>
<td>f</td>
<td>1e-15</td>
</tr>
<tr>
<td>atto</td>
<td>a</td>
<td>1e-18</td>
</tr>
<tr>
<td>zepto</td>
<td>z</td>
<td>1e-21</td>
</tr>
<tr>
<td>yocto</td>
<td>y</td>
<td>1e-23</td>
</tr>
</tbody>
</table>

Table 1: SI prefixes [2, 121]
4.2 Base Units

The lua-physical library has nine base quantities. These are the seven basis units or basis quantities of the SI system \[3\] and in addition the base quantity of information \_bit\ and of currency \_EUR\. All other quantities are derived from these base units.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Dim.</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>–</td>
<td>_1</td>
<td>1</td>
<td>The dimensionless number one.</td>
</tr>
<tr>
<td>time</td>
<td>second</td>
<td>_s</td>
<td>T</td>
<td>The SI unit of time. It is defined by taking the fixed numerical value of the caesium frequency (\Delta\nu_{Cs}), the unperturbed ground-state hyperfine transition frequency of the caesium 133 atom, to be 9192631.770 when expressed in the unit (1/s).</td>
</tr>
<tr>
<td>length</td>
<td>meter</td>
<td>_m</td>
<td>L</td>
<td>The SI unit of length. It is defined by taking the fixed numerical value of the speed of light in vacuum (c) to be 299792.458 when expressed in the unit (m/s).</td>
</tr>
</tbody>
</table>

\[\text{The number one is a unit with dimension zero. Strictly speaking it is not a base unit.}\]
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Dim.</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass</td>
<td>kilogram</td>
<td>kg</td>
<td>M</td>
<td>The SI unit of mass. It is defined by taking the fixed numerical value of the Planck constant $h$ to be $6.62607015 \times 10^{-34}$ when expressed in m$^2$kg/s.</td>
</tr>
<tr>
<td>electric current</td>
<td>ampere</td>
<td>A</td>
<td>I</td>
<td>The SI unit of electric current. It is defined by taking the fixed numerical value of the elementary charge $e$ to be $1.602176634 \times 10^{-19}$ when expressed in A·s.</td>
</tr>
<tr>
<td>thermodynamic</td>
<td>kelvin</td>
<td>K</td>
<td>K$^1$</td>
<td>The SI unit of the thermodynamic temperature. It is defined by taking the fixed numerical value of the Boltzmann constant $k_B$ to be $1.380649 \times 10^{-23}$ when expressed in kg·m$^2$/s$^2$K.</td>
</tr>
<tr>
<td>amount of substance</td>
<td>mole</td>
<td>mol</td>
<td>N</td>
<td>The SI unit of amount of substance. One mole contains exactly $6.02214076 \times 10^{23}$ elementary entities. This number is the fixed numerical value of the Avogadro constant $N_A$ when expressed in 1/mol.</td>
</tr>
<tr>
<td>luminous intensity</td>
<td>candela</td>
<td>cd</td>
<td>J</td>
<td>The SI unit of luminous intensity in a given direction. It is defined by taking the fixed numerical value of the luminous efficacy of monochromatic radiation of frequency $5.4 \times 10^{14}$ Hz, $K_{cd}$, to be 683 when expressed in the unit cd·sr·s$^3$/(kg·m$^2$).</td>
</tr>
<tr>
<td>information</td>
<td>bit</td>
<td>bit</td>
<td>B</td>
<td>The smallest amount of information.</td>
</tr>
<tr>
<td>currency</td>
<td>euro</td>
<td>EUR</td>
<td>C</td>
<td>The value of the currency Euro.</td>
</tr>
</tbody>
</table>

$^1$The SI symbol for the dimension of temperature is $\Theta$, but all symbols of this library consist of roman letters, numbers and underscores only. Therefore the symbol for the dimension of the thermodynamic temperature is the letter K.
### 4.3 Constants

All physical constants are taken from the NIST webpage [1].

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi</td>
<td><em>π</em></td>
<td>3.1415926535897932384626433832795028841971 * _1</td>
</tr>
<tr>
<td>eulersnumber</td>
<td><em>e</em></td>
<td>2.7182818284590452353602874713526624977572 * _1</td>
</tr>
<tr>
<td>speedoflight</td>
<td><em>c</em></td>
<td>299792458 * _m/_s</td>
</tr>
<tr>
<td>gravitationalconstant</td>
<td><em>Gc</em></td>
<td>N(6.67408e-11,3.1e-15) * _m^3/(_kg* _s^2)</td>
</tr>
<tr>
<td>planckconstant</td>
<td><em>h_P</em></td>
<td>6.62607015e-34 * _J*_s</td>
</tr>
<tr>
<td>reducedplanckconstant</td>
<td><em>h_Pbar</em></td>
<td>_h_P/(2*_π)</td>
</tr>
<tr>
<td>elementarycharge</td>
<td><em>e</em></td>
<td>1.602176634e-19 * _C</td>
</tr>
<tr>
<td>vacuumpermeability</td>
<td>_μ_0</td>
<td>4e-7*π * _N/_A^2</td>
</tr>
<tr>
<td>vacuumpermittivity</td>
<td>_ε_0</td>
<td>1/((_μ_0* _c^2))</td>
</tr>
<tr>
<td>atomicmassunit</td>
<td><em>u</em></td>
<td>N(1.66053904e-27,2e-35) * _kg</td>
</tr>
<tr>
<td>electronmass</td>
<td>_m_e</td>
<td>N(9.10938356e-31,1.1e-38) * _kg</td>
</tr>
<tr>
<td>protonmass</td>
<td>_m_p</td>
<td>N(1.672621898e-27,2.1e-35) * _kg</td>
</tr>
<tr>
<td>neutronmass</td>
<td>_m_n</td>
<td>N(1.674927471e-27,2.1e-35) * _kg</td>
</tr>
<tr>
<td>bohrmagneton</td>
<td><em>μ_B</em></td>
<td>_e* _h_Pbar/(2*_m_e)</td>
</tr>
<tr>
<td>nuclearmagneton</td>
<td><em>μ_N</em></td>
<td>_e* _h_Pbar/(2*_m_p)</td>
</tr>
<tr>
<td>electromagneticmoment</td>
<td><em>μ_e</em></td>
<td>N(-928.4764620e-26,5.7e-32) * _J/_T</td>
</tr>
<tr>
<td>protonmagneticmoment</td>
<td><em>μ_p</em></td>
<td>N(1.4106067873e-26,9.7e-35) * _J/_T</td>
</tr>
<tr>
<td>neutronmagneticmoment</td>
<td><em>μ_n</em></td>
<td>N(-0.96623650e-26,2.3e-26) * _J/_T</td>
</tr>
<tr>
<td>finestructureconstant</td>
<td><em>α</em></td>
<td>_μ_0* _e^2<em>_c/(2</em>_h_P)</td>
</tr>
<tr>
<td>rydbergconstant</td>
<td><em>Ry</em></td>
<td>_α^2 _m_e<em>_c/(2</em>_h_P)</td>
</tr>
<tr>
<td>avogadronumber</td>
<td><em>N_A</em></td>
<td>6.02214076e23/_mol</td>
</tr>
<tr>
<td>boltzmannconstant</td>
<td><em>k_B</em></td>
<td>1.380649e-23 * _J/_K</td>
</tr>
<tr>
<td>molargasconstant</td>
<td><em>R</em></td>
<td>N(8.3144598,4.8e-6) * _J/(_K* _mol)</td>
</tr>
<tr>
<td>stefanboltzmannconstant</td>
<td><em>σ</em></td>
<td>_π^2 _k_B^4/(60<em>_h_Pbar^3</em>_c^2)</td>
</tr>
<tr>
<td>standardgravity</td>
<td>_g_0</td>
<td>9.80665 * _m/ _s^2</td>
</tr>
</tbody>
</table>

Table 4: Physical and mathematical constants
## 4.4 Coherent derived units in the SI

All units in this section are coherent derived units from the SI base units with special names, \cite{[2] 118}.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane Angle</td>
<td>radian</td>
<td>_rad</td>
<td>1</td>
</tr>
<tr>
<td>Solid Angle</td>
<td>steradian</td>
<td>_sr</td>
<td>_rad^2</td>
</tr>
<tr>
<td>Frequency</td>
<td>hertz</td>
<td>_Hz</td>
<td>1/_s</td>
</tr>
<tr>
<td>Force</td>
<td>newton</td>
<td>_N</td>
<td>_kg*_m/_s^2</td>
</tr>
<tr>
<td>Pressure</td>
<td>pascal</td>
<td>_Pa</td>
<td>_N/_m^2</td>
</tr>
<tr>
<td>Energy</td>
<td>joule</td>
<td>_J</td>
<td>_N*_m</td>
</tr>
<tr>
<td>Power</td>
<td>watt</td>
<td>_W</td>
<td>_J/_s</td>
</tr>
<tr>
<td>Electric Charge</td>
<td>coulomb</td>
<td>_C</td>
<td>_A*_s</td>
</tr>
<tr>
<td>Electric Potential</td>
<td>volt</td>
<td>_V</td>
<td>_J/_C</td>
</tr>
<tr>
<td>Electric Capacitance</td>
<td>farad</td>
<td>_F</td>
<td>_C/_V</td>
</tr>
<tr>
<td>Electric Resistance</td>
<td>ohm</td>
<td>_Ohm</td>
<td>_V/_A</td>
</tr>
<tr>
<td>Electric Conductance</td>
<td>siemens</td>
<td>_S</td>
<td>_A/_V</td>
</tr>
<tr>
<td>Magnetic Flux</td>
<td>weber</td>
<td>_Wb</td>
<td>_V*_s</td>
</tr>
<tr>
<td>Magnetic Flux Density</td>
<td>tesla</td>
<td>_T</td>
<td>_Wb/_m^2</td>
</tr>
<tr>
<td>Inductance</td>
<td>henry</td>
<td>_H</td>
<td>_Wb/_A</td>
</tr>
<tr>
<td>Temperature</td>
<td>celsius</td>
<td>_degC</td>
<td>_K</td>
</tr>
<tr>
<td>Luminous Flux</td>
<td>lumen</td>
<td>_lm</td>
<td>_cd*_sr</td>
</tr>
<tr>
<td>Illuminance</td>
<td>lux</td>
<td>_lx</td>
<td>_lm/_m^2</td>
</tr>
<tr>
<td>Activity</td>
<td>becquerel</td>
<td>_Bq</td>
<td>1/_s</td>
</tr>
<tr>
<td>Absorbed Dose</td>
<td>gray</td>
<td>_Gy</td>
<td>_J/_kg</td>
</tr>
<tr>
<td>Dose Equivalent</td>
<td>sievert</td>
<td>_Sv</td>
<td>_J/_kg</td>
</tr>
<tr>
<td>Catalytic Activity</td>
<td>katal</td>
<td>_kat</td>
<td>_mol/_s</td>
</tr>
</tbody>
</table>

1In the SI system, the quantity Plane Angle has the dimension of a number.
2In the SI system, the quantity Solid Angle has the dimension of a number.
3The unit \_PS stands for peta siemens and is in conflict with the metric version of the unit horsepower (German Pferdestärke). Since the latter is more common than peta siemens, \_PS is defined to be the metric version of horsepower.
4The unit \_degC is by default interpreted as a temperature difference.
4.5 Non-SI units accepted for use with the SI

There are a few units with dimension 1. \[2, 124\].

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>minute</td>
<td>_min</td>
<td>60 * _s</td>
</tr>
<tr>
<td></td>
<td>hour</td>
<td>_h</td>
<td>60 * _min</td>
</tr>
<tr>
<td></td>
<td>day</td>
<td>_d</td>
<td>24 * _h</td>
</tr>
<tr>
<td>Plane Angle</td>
<td>degree</td>
<td>_deg</td>
<td>((\pi/180) * _rad)</td>
</tr>
<tr>
<td></td>
<td>arcminute</td>
<td>_arcmin</td>
<td>_deg/60</td>
</tr>
<tr>
<td></td>
<td>arcsecond</td>
<td>_arcsec</td>
<td>_arcmin/60</td>
</tr>
<tr>
<td>Area</td>
<td>hectare</td>
<td>_hectare</td>
<td>1e4 * _m^2</td>
</tr>
<tr>
<td>Volume</td>
<td>liter</td>
<td>_L</td>
<td>1e-3 * _m^3</td>
</tr>
<tr>
<td>Mass</td>
<td>tonne</td>
<td>_t</td>
<td>1e3 * _kg</td>
</tr>
</tbody>
</table>

4.6 Nominal Astronomical Units

The nominal values of solar, terrestrial and jovial quantities are taken from IAU Resolution B3 \[3\].

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>nomsolradius</td>
<td>_R_S_nom</td>
<td>6.957e8 * _m</td>
</tr>
<tr>
<td>Irradiance</td>
<td>nomsolirradiance</td>
<td>_S_S_nom</td>
<td>1361 * _W/_m^2</td>
</tr>
<tr>
<td>Radiant Flux</td>
<td>nomsolluminosity</td>
<td>_L_S_nom</td>
<td>3.828e26 * _W</td>
</tr>
<tr>
<td>Temperature</td>
<td>nomsolefftemperature</td>
<td>_T_S_nom</td>
<td>5772 * _K</td>
</tr>
<tr>
<td>Mass Parameter</td>
<td>nomsolmassparameter</td>
<td>_GM_S_nom</td>
<td>1.3271244e20 * _m^3 * _s^-2</td>
</tr>
<tr>
<td>Length</td>
<td>nomterreqradius</td>
<td>_Re_E_nom</td>
<td>6.3781e6 * _m</td>
</tr>
<tr>
<td>Length</td>
<td>nomterrpolaradius</td>
<td>_Rp_E_nom</td>
<td>6.3568e6 * _m</td>
</tr>
<tr>
<td>Mass Parameter</td>
<td>nomterrmassparameter</td>
<td>_GM_E_nom</td>
<td>3.986004e14 * _m^3 * _s^-2</td>
</tr>
<tr>
<td>Length</td>
<td>nomjoveqradius</td>
<td>_Re_J_nom</td>
<td>7.1492e7 * _m</td>
</tr>
<tr>
<td>Length</td>
<td>nomjovpolaradius</td>
<td>_Rp_J_nom</td>
<td>6.6854e7 * _m</td>
</tr>
<tr>
<td>Mass Parameter</td>
<td>nomjovmassparameter</td>
<td>_GM_J_nom</td>
<td>1.2668653e17 * _m^3 * _s^-2</td>
</tr>
</tbody>
</table>
### 4.7 Other Non-SI units

The unit Bel is only available with prefix decibel, because \( B \) is the unit byte.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>angstrom</td>
<td>_angstrom</td>
<td>1e-10 * _m</td>
</tr>
<tr>
<td></td>
<td>fermi</td>
<td>_fermi</td>
<td>1e-15 * _m</td>
</tr>
<tr>
<td>Time</td>
<td>svedberg</td>
<td>_svedberg</td>
<td>1e-13 * _s</td>
</tr>
<tr>
<td></td>
<td>week</td>
<td>_wk</td>
<td>7 * _d</td>
</tr>
<tr>
<td></td>
<td>year</td>
<td>_a</td>
<td>365.25 * _d</td>
</tr>
<tr>
<td></td>
<td>astronomicalunit</td>
<td>_au</td>
<td>149597870700 * _m</td>
</tr>
<tr>
<td></td>
<td>lightsecond</td>
<td>_ls</td>
<td>_c*_s</td>
</tr>
<tr>
<td></td>
<td>lightyear</td>
<td>_ly</td>
<td>_c*_a</td>
</tr>
<tr>
<td></td>
<td>parsec</td>
<td>_pc</td>
<td>(648000/_Pi) * _au</td>
</tr>
<tr>
<td>Area</td>
<td>barn</td>
<td>_barn</td>
<td>1e-28 * _m^2</td>
</tr>
<tr>
<td></td>
<td>are</td>
<td>_are</td>
<td>1e2 * _m^2</td>
</tr>
<tr>
<td>Volume</td>
<td>metricteaspoon</td>
<td>_tsp</td>
<td>5e-3 * _L</td>
</tr>
<tr>
<td></td>
<td>metrictablespoon</td>
<td>_Tbsp</td>
<td>3 * _tsp</td>
</tr>
<tr>
<td>Plane Angle</td>
<td>gradian</td>
<td>_gon</td>
<td>(Pi/200) * _rad</td>
</tr>
<tr>
<td></td>
<td>turn</td>
<td>_tr</td>
<td>2*Pi * _rad</td>
</tr>
<tr>
<td>Solid Angle</td>
<td>spat</td>
<td>_sp</td>
<td>4*Pi * _sr</td>
</tr>
<tr>
<td>Force</td>
<td>kilopond</td>
<td>_kp</td>
<td>_kg*_g_0</td>
</tr>
<tr>
<td>Pressure</td>
<td>bar</td>
<td>_bar</td>
<td>1e5 * _Pa</td>
</tr>
<tr>
<td></td>
<td>standardatmosphere</td>
<td>_atm</td>
<td>101325 * _Pa</td>
</tr>
<tr>
<td></td>
<td>technicalatmosphere</td>
<td>_at</td>
<td>_kp/_cm^2</td>
</tr>
<tr>
<td></td>
<td>millimeterofmercury</td>
<td>_mmHg</td>
<td>133.322387415 * _Pa</td>
</tr>
<tr>
<td></td>
<td>torr</td>
<td>_Torr</td>
<td>(101325/760) * _Pa</td>
</tr>
<tr>
<td>Energy</td>
<td>thermochemicalcalorie</td>
<td>_cal</td>
<td>4.184 * _J</td>
</tr>
<tr>
<td></td>
<td>internationalcalorie</td>
<td>_cal_IT</td>
<td>4.1868 * _J</td>
</tr>
<tr>
<td></td>
<td>gramoftnt</td>
<td>_g_TNT</td>
<td>1e3 * _cal</td>
</tr>
<tr>
<td></td>
<td>tonofnt</td>
<td>_t_TNT</td>
<td>1e9 * _cal</td>
</tr>
<tr>
<td>Quantity</td>
<td>Unit</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>electronvolt</td>
<td>_eV</td>
<td>_e*V</td>
<td></td>
</tr>
<tr>
<td>wattsecond</td>
<td>_Ws</td>
<td>_W*s</td>
<td></td>
</tr>
<tr>
<td>watthour</td>
<td>_Wh</td>
<td>_W*h</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>voltampere</td>
<td>_VA</td>
<td>_V*A</td>
</tr>
<tr>
<td>Electric Charge</td>
<td>amperesecond</td>
<td>_As</td>
<td>_A*s</td>
</tr>
<tr>
<td></td>
<td>amperehour</td>
<td>_Ah</td>
<td>_A*h</td>
</tr>
<tr>
<td>Information</td>
<td>nibble</td>
<td>_nibble</td>
<td>4 * _bit</td>
</tr>
<tr>
<td></td>
<td>byte</td>
<td>_B</td>
<td>8 * _bit</td>
</tr>
<tr>
<td>Information</td>
<td>bitpersecond</td>
<td>_bps</td>
<td>_bit/s</td>
</tr>
<tr>
<td>Transfer Rate</td>
<td>percent</td>
<td>_.percent</td>
<td>1e-2 * _1</td>
</tr>
<tr>
<td></td>
<td>permille</td>
<td>_.permille</td>
<td>1e-3 * _1</td>
</tr>
<tr>
<td></td>
<td>partspermillion</td>
<td>_.ppm</td>
<td>1e-6 * _1</td>
</tr>
<tr>
<td></td>
<td>partsperbillion</td>
<td>_.ppb</td>
<td>1e-9 * _1</td>
</tr>
<tr>
<td></td>
<td>partspertrillion</td>
<td>_.ppt</td>
<td>1e-12 * _1</td>
</tr>
<tr>
<td></td>
<td>partsperquadrillion</td>
<td>_.ppq</td>
<td>1e-15 * _1</td>
</tr>
<tr>
<td></td>
<td>decibel</td>
<td>_.dB</td>
<td>_1</td>
</tr>
<tr>
<td>Power</td>
<td>metrichorsepower</td>
<td>_PS</td>
<td>75 * _g_0* _kg*_m/_s</td>
</tr>
<tr>
<td>Activity</td>
<td>curie</td>
<td>_Ci</td>
<td>3.7e10 * _Bq</td>
</tr>
<tr>
<td>Absorbed Dose</td>
<td>rad</td>
<td>_Rad</td>
<td>1e-2 * _Gy</td>
</tr>
<tr>
<td>Dose Equivalent</td>
<td>rem</td>
<td>_rem</td>
<td>1e-2 * _Sv</td>
</tr>
<tr>
<td>Viscosity</td>
<td>poiseuille</td>
<td>_Pl</td>
<td>_Pa*s</td>
</tr>
</tbody>
</table>
### 4.8 Imperial Units

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>inch</td>
<td>_in</td>
<td>2.54e-2 * _m</td>
</tr>
<tr>
<td></td>
<td>thou</td>
<td>_th</td>
<td>1e-3 * _in</td>
</tr>
<tr>
<td>DTP Point</td>
<td>point</td>
<td>_pt</td>
<td>_in/72</td>
</tr>
<tr>
<td></td>
<td>pica</td>
<td>_pica</td>
<td>12 * _pt</td>
</tr>
<tr>
<td></td>
<td>hand</td>
<td>_hh</td>
<td>4 * _in</td>
</tr>
<tr>
<td></td>
<td>foot</td>
<td>_ft</td>
<td>12 * _in</td>
</tr>
<tr>
<td></td>
<td>yard</td>
<td>_yd</td>
<td>3 * _ft</td>
</tr>
<tr>
<td></td>
<td>rod</td>
<td>_rd</td>
<td>5.5 * _yd</td>
</tr>
<tr>
<td></td>
<td>chain</td>
<td>_ch</td>
<td>4 * _rd</td>
</tr>
<tr>
<td></td>
<td>furlong</td>
<td>_fur</td>
<td>10 * _ch</td>
</tr>
<tr>
<td></td>
<td>mile</td>
<td>_mi</td>
<td>8 * _fur</td>
</tr>
<tr>
<td></td>
<td>league</td>
<td>_lea</td>
<td>3*_mi</td>
</tr>
<tr>
<td></td>
<td>nauticalmile</td>
<td>_nmi</td>
<td>1852 * _m</td>
</tr>
<tr>
<td></td>
<td>nauticalleague</td>
<td>_nlea</td>
<td>3 * _nmi</td>
</tr>
<tr>
<td></td>
<td>cable</td>
<td>_cbl</td>
<td>0.1 * _nmi</td>
</tr>
<tr>
<td></td>
<td>fathom</td>
<td>_ftm</td>
<td>6 * _ft</td>
</tr>
<tr>
<td>Velocity</td>
<td>knot</td>
<td>_kn</td>
<td>_nmi/_h</td>
</tr>
<tr>
<td>Area</td>
<td>acre</td>
<td>_ac</td>
<td>10 * _ch^2</td>
</tr>
<tr>
<td>Volume</td>
<td>gallon</td>
<td>_gal</td>
<td>4.54609* _L</td>
</tr>
<tr>
<td></td>
<td>quart</td>
<td>_qt</td>
<td>_gal/4</td>
</tr>
<tr>
<td></td>
<td>pint</td>
<td>_pint</td>
<td>_qt/2</td>
</tr>
<tr>
<td></td>
<td>cup</td>
<td>_cup</td>
<td>_pint/2</td>
</tr>
<tr>
<td></td>
<td>gill</td>
<td>_gi</td>
<td>_pint/4</td>
</tr>
<tr>
<td></td>
<td>fluidounce</td>
<td>_fl_oz</td>
<td>_gi/5</td>
</tr>
<tr>
<td></td>
<td>fluid dram</td>
<td>_fl_dr</td>
<td>_fl_oz/8</td>
</tr>
</tbody>
</table>

1 The desktop publishing point or PostScript point is 1/72 of an international inch.
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>grain</td>
<td>_gr</td>
<td>64.79891*_mg</td>
</tr>
<tr>
<td></td>
<td>pound</td>
<td>_lb</td>
<td>7000*_gr</td>
</tr>
<tr>
<td></td>
<td>ounce</td>
<td>_oz</td>
<td>_lb/16</td>
</tr>
<tr>
<td></td>
<td>dram</td>
<td>_dr</td>
<td>_lb/256</td>
</tr>
<tr>
<td></td>
<td>stone</td>
<td>_st</td>
<td>14*_lb</td>
</tr>
<tr>
<td></td>
<td>quarter</td>
<td>_qtr</td>
<td>2*_st</td>
</tr>
<tr>
<td></td>
<td>hundredweight</td>
<td>_cwt</td>
<td>4*_qtr</td>
</tr>
<tr>
<td></td>
<td>longton</td>
<td>_ton</td>
<td>20*_cwt</td>
</tr>
<tr>
<td></td>
<td>troy pound</td>
<td>_lb_t</td>
<td>5760*_gr</td>
</tr>
<tr>
<td></td>
<td>troy ounce</td>
<td>_oz_t</td>
<td>_lb_t/12</td>
</tr>
<tr>
<td></td>
<td>pennyweight</td>
<td>_dwt</td>
<td>24*_gr</td>
</tr>
<tr>
<td></td>
<td>firkin</td>
<td>_fir</td>
<td>56*_lb</td>
</tr>
<tr>
<td>Time</td>
<td>sennight</td>
<td>_sen</td>
<td>7*_d</td>
</tr>
<tr>
<td></td>
<td>fort night</td>
<td>_ftn</td>
<td>14*_d</td>
</tr>
<tr>
<td>Temperature</td>
<td>fahrenheit</td>
<td>_degF</td>
<td>(5/9)*_K</td>
</tr>
<tr>
<td>Force</td>
<td>poundforce</td>
<td>_lbf</td>
<td>_lb*_g_0</td>
</tr>
<tr>
<td></td>
<td>poundal</td>
<td>_pdl</td>
<td>_lb*_ft/_s^2</td>
</tr>
<tr>
<td>Mass</td>
<td>slug</td>
<td>_slug</td>
<td>_lbf*_s^2/_ft</td>
</tr>
<tr>
<td>Pressure</td>
<td>poundforcepersquareinch</td>
<td>_psi</td>
<td>_lbf/in^2</td>
</tr>
<tr>
<td>Torque,Energy</td>
<td>thcmbritishthermalunit</td>
<td>_BTU</td>
<td>(1897.83047608/1.8)*_J</td>
</tr>
<tr>
<td></td>
<td>intbritishthermalunit</td>
<td>_BTU_it</td>
<td>1055.05585262 * _J</td>
</tr>
<tr>
<td>Power</td>
<td>horsepower</td>
<td>_hp</td>
<td>33000<em>_ft</em>_lbf/_min</td>
</tr>
</tbody>
</table>

1 The unit _degF is by default interpreted as a temperature difference.
4.9 U.S. customary units

In the U.S., the length units are bound to the meter differently than in the imperial system. The following definitions are taken from [https://en.wikipedia.org/wiki/United_States_customary_units](https://en.wikipedia.org/wiki/United_States_customary_units).

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>ussurveyinch</td>
<td>_in_US</td>
<td>( \frac{\text{m}}{39.37} )</td>
</tr>
<tr>
<td></td>
<td>ussurveyhand</td>
<td>_hh_US</td>
<td>4 * _in_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyfoot</td>
<td>_ft_US</td>
<td>3 * _hh_US</td>
</tr>
<tr>
<td></td>
<td>ussurveylink</td>
<td>_li_US</td>
<td>0.66 * _ft_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyyard</td>
<td>_yd_US</td>
<td>3 * _ft_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyrod</td>
<td>_rd_US</td>
<td>5.5 * _yd_US</td>
</tr>
<tr>
<td></td>
<td>ussurveychain</td>
<td>_ch_US</td>
<td>4 * _rd_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyfurlong</td>
<td>_fur_US</td>
<td>10 * _ch_US</td>
</tr>
<tr>
<td></td>
<td>ussurveymile</td>
<td>_mi_US</td>
<td>8 * _fur_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyleague</td>
<td>_lea_US</td>
<td>3 * _mi_US</td>
</tr>
<tr>
<td></td>
<td>ussurveyfathom</td>
<td>_ftm_US</td>
<td>72 * _in_US</td>
</tr>
<tr>
<td></td>
<td>ussurveycable</td>
<td>_cbl_US</td>
<td>120 * _ftm_US</td>
</tr>
<tr>
<td>Area</td>
<td>ussurveyacre</td>
<td>_ac_US</td>
<td>_ch_US * _fur_US</td>
</tr>
<tr>
<td>Volume</td>
<td>usgallon</td>
<td>_gal_US</td>
<td>231 * _in^3</td>
</tr>
<tr>
<td></td>
<td>usquart</td>
<td>_qt_US</td>
<td>_gal_US/4</td>
</tr>
<tr>
<td></td>
<td>uspint</td>
<td>_pint_US</td>
<td>_qt_US/2</td>
</tr>
<tr>
<td></td>
<td>uscup</td>
<td>_cup_US</td>
<td>_pint_US/2</td>
</tr>
<tr>
<td></td>
<td>usgill</td>
<td>_gi_US</td>
<td>_pint_US/4</td>
</tr>
<tr>
<td></td>
<td>usfluidounce</td>
<td>_fl_oz_US</td>
<td>_gi_US/4</td>
</tr>
<tr>
<td></td>
<td>устоловушка</td>
<td>_Tbsp_US</td>
<td>_fl_oz_US/2</td>
</tr>
<tr>
<td></td>
<td>устеаспун</td>
<td>_tsp_US</td>
<td>_Tbsp_US/3</td>
</tr>
<tr>
<td></td>
<td>ustfluiddram</td>
<td>_fl_dr_US</td>
<td>_fl_oz_US/8</td>
</tr>
<tr>
<td>Mass</td>
<td>usquarter</td>
<td>_qtr_US</td>
<td>25 * _lb</td>
</tr>
<tr>
<td></td>
<td>ushundredweight</td>
<td>_cwt_US</td>
<td>4 * _qtr_US</td>
</tr>
<tr>
<td></td>
<td>usaton</td>
<td>_ton_US</td>
<td>20 * _cwt_US</td>
</tr>
</tbody>
</table>

22
### 4.10 International Currencies

International currency units based on exchange rates from 7.3.2019, 21:00 UTC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghan Afghani</td>
<td>_AFN</td>
<td>0.012 * _EUR</td>
</tr>
<tr>
<td>Afghan Pul</td>
<td>_cAFN</td>
<td>0.01 * _AFN</td>
</tr>
<tr>
<td>Albanian Lek</td>
<td>_ALL</td>
<td>0.008 * _EUR</td>
</tr>
<tr>
<td>Armenian Dram</td>
<td>_AMD</td>
<td>0.0018 * _EUR</td>
</tr>
<tr>
<td>Armenian Luma</td>
<td>_cAMD</td>
<td>0.01 * _AMD</td>
</tr>
<tr>
<td>Angolan Kwanza</td>
<td>_AOA</td>
<td>0.0028 * _EUR</td>
</tr>
<tr>
<td>Angolan Centimo</td>
<td>_cAOA</td>
<td>0.01 * _AOA</td>
</tr>
<tr>
<td>Argentine Peso</td>
<td>_ARS</td>
<td>0.021 * _EUR</td>
</tr>
<tr>
<td>Argentine Centavo</td>
<td>_cARS</td>
<td>0.01 * _ARS</td>
</tr>
<tr>
<td>Australian Dollar</td>
<td>_AUD</td>
<td>0.63 * _EUR</td>
</tr>
<tr>
<td>Australian Cent</td>
<td>_cAUD</td>
<td>0.01 * _AUD</td>
</tr>
<tr>
<td>Azerbaijani Manat</td>
<td>_AZN</td>
<td>0.63 * _EUR</td>
</tr>
<tr>
<td>Azerbaijani Qepik</td>
<td>_cAZN</td>
<td>0.01 * _AZN</td>
</tr>
<tr>
<td>Bosnian Mark</td>
<td>_BAM</td>
<td>0.51 * _EUR</td>
</tr>
<tr>
<td>Bosnian Fenings</td>
<td>_cBAM</td>
<td>0.01 * _BAM</td>
</tr>
<tr>
<td>Bangladeshi Taka</td>
<td>_BDT</td>
<td>0.011 * _EUR</td>
</tr>
<tr>
<td>Bangladeshi Poisha</td>
<td>_cBDT</td>
<td>0.01 * _BDT</td>
</tr>
<tr>
<td>Burundian Franc</td>
<td>_BIF</td>
<td>0.00049 * _EUR</td>
</tr>
<tr>
<td>Burundian Centime</td>
<td>_cBIF</td>
<td>0.01 * _BIF</td>
</tr>
<tr>
<td>Bolivian Boliviano</td>
<td>_BOB</td>
<td>0.13 * _EUR</td>
</tr>
<tr>
<td>Bolivian Centavo</td>
<td>_cBOB</td>
<td>0.01 * _BOB</td>
</tr>
<tr>
<td>Brazilian Real</td>
<td>_BRL</td>
<td>0.23 * _EUR</td>
</tr>
<tr>
<td>Brazilian Centavo</td>
<td>_cBRL</td>
<td>0.01 * _BRL</td>
</tr>
<tr>
<td>Botswana Pula</td>
<td>_BWP</td>
<td>0.083 * _EUR</td>
</tr>
<tr>
<td>Botswana Thebe</td>
<td>_cBWP</td>
<td>0.01 * _BWP</td>
</tr>
<tr>
<td>Belarusian Ruble</td>
<td>_BYN</td>
<td>0.42 * _EUR</td>
</tr>
<tr>
<td>Belarusian Kapiecja</td>
<td>_cBYN</td>
<td>0.01 * _BYN</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CanadianDollar</td>
<td>_CAD</td>
<td>0.66 * _EUR</td>
</tr>
<tr>
<td>CanadianCent</td>
<td>_cCAD</td>
<td>0.01 * _CAD</td>
</tr>
<tr>
<td>CongoleseFranc</td>
<td>_CDF</td>
<td>0.00055 * _EUR</td>
</tr>
<tr>
<td>CongoleseCentime</td>
<td>_cCDF</td>
<td>0.01 * _CDF</td>
</tr>
<tr>
<td>SwissFranc</td>
<td>_CHF</td>
<td>0.88 * _EUR</td>
</tr>
<tr>
<td>SwissRappen</td>
<td>_cCHF</td>
<td>0.01 * _CHF</td>
</tr>
<tr>
<td>ChileanPeso</td>
<td>_CLP</td>
<td>0.0013 * _EUR</td>
</tr>
<tr>
<td>ChileanCentavo</td>
<td>_cCLP</td>
<td>0.01 * _CLP</td>
</tr>
<tr>
<td>ChineseRenminbiYuan</td>
<td>_CNY</td>
<td>0.13 * _EUR</td>
</tr>
<tr>
<td>ChineseRenminbiFen</td>
<td>_cCNY</td>
<td>0.01 * _CNY</td>
</tr>
<tr>
<td>ColombianPeso</td>
<td>_COP</td>
<td>0.00028 * _EUR</td>
</tr>
<tr>
<td>ColombianCentavo</td>
<td>_cCOP</td>
<td>0.01 * _COP</td>
</tr>
<tr>
<td>CostaRicanColon</td>
<td>_CRC</td>
<td>0.0015 * _EUR</td>
</tr>
<tr>
<td>CostaRicanCentimos</td>
<td>_cCRC</td>
<td>0.01 * _CRC</td>
</tr>
<tr>
<td>CzechKoruna</td>
<td>_CZK</td>
<td>0.039 * _EUR</td>
</tr>
<tr>
<td>CzechHaler</td>
<td>_cCZK</td>
<td>0.01 * _CZK</td>
</tr>
<tr>
<td>DanishKrone</td>
<td>_DKK</td>
<td>0.13 * _EUR</td>
</tr>
<tr>
<td>DanishOre</td>
<td>_cDKK</td>
<td>0.01 * _DKK</td>
</tr>
<tr>
<td>DominicanPeso</td>
<td>_DOP</td>
<td>0.018 * _EUR</td>
</tr>
<tr>
<td>DominicanCentavo</td>
<td>_cDOP</td>
<td>0.01 * _DOP</td>
</tr>
<tr>
<td>AlgerianDinar</td>
<td>_DZD</td>
<td>0.0074 * _EUR</td>
</tr>
<tr>
<td>AlgerianSanteem</td>
<td>_cDZD</td>
<td>0.01 * _DZD</td>
</tr>
<tr>
<td>EgyptianPound</td>
<td>_EGP</td>
<td>0.051 * _EUR</td>
</tr>
<tr>
<td>EgyptianPiastre</td>
<td>_cEGP</td>
<td>0.01 * _EGP</td>
</tr>
<tr>
<td>EthiopianBirr</td>
<td>_ETB</td>
<td>0.031 * _EUR</td>
</tr>
<tr>
<td>EthiopianSantim</td>
<td>_cETB</td>
<td>0.01 * _ETB</td>
</tr>
<tr>
<td>FijianDollar</td>
<td>_FJD</td>
<td>0.42 * _EUR</td>
</tr>
<tr>
<td>FijianCent</td>
<td>_cFJD</td>
<td>0.01 * _FJD</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>----------------</td>
</tr>
<tr>
<td>PoundSterling</td>
<td>_GBP</td>
<td>1.16 * _EUR</td>
</tr>
<tr>
<td>PennySterling</td>
<td>_cGBP</td>
<td>0.01 * _GBP</td>
</tr>
<tr>
<td>GeorgianLari</td>
<td>_GEL</td>
<td>0.33 * _EUR</td>
</tr>
<tr>
<td>GeorgianTetri</td>
<td>_cGEL</td>
<td>0.01 * _GEL</td>
</tr>
<tr>
<td>GhanaianCedi</td>
<td>_GHS</td>
<td>0.16 * _EUR</td>
</tr>
<tr>
<td>GhanaianPresewa</td>
<td>_cGHS</td>
<td>0.01 * _GHS</td>
</tr>
<tr>
<td>GambianDalasi</td>
<td>_GMD</td>
<td>0.018 * _EUR</td>
</tr>
<tr>
<td>GambianButut</td>
<td>_cGMD</td>
<td>0.01 * _GMD</td>
</tr>
<tr>
<td>GuineanFranc</td>
<td>_GNF</td>
<td>9.6e-05 * _EUR</td>
</tr>
<tr>
<td>GuineanCentime</td>
<td>_cGNF</td>
<td>0.01 * _GNF</td>
</tr>
<tr>
<td>GuatemalanQuetzal</td>
<td>_GTQ</td>
<td>0.12 * _EUR</td>
</tr>
<tr>
<td>GuatemalanCentavo</td>
<td>_cGTQ</td>
<td>0.01 * _GTQ</td>
</tr>
<tr>
<td>GuyaneseDollar</td>
<td>_GYD</td>
<td>0.0043 * _EUR</td>
</tr>
<tr>
<td>GuyaneseCent</td>
<td>_cGYD</td>
<td>0.01 * _GYD</td>
</tr>
<tr>
<td>HongKongDollar</td>
<td>_HKD</td>
<td>0.11 * _EUR</td>
</tr>
<tr>
<td>HongKongCent</td>
<td>_cHKD</td>
<td>0.01 * _HKD</td>
</tr>
<tr>
<td>HonduranLempira</td>
<td>_HNL</td>
<td>0.036 * _EUR</td>
</tr>
<tr>
<td>HonduranCentavo</td>
<td>_cHNL</td>
<td>0.01 * _HNL</td>
</tr>
<tr>
<td>CroatianKuna</td>
<td>_HRK</td>
<td>0.13 * _EUR</td>
</tr>
<tr>
<td>CroatianLipa</td>
<td>_cHRK</td>
<td>0.01 * _HRK</td>
</tr>
<tr>
<td>HaitianGourde</td>
<td>_HTG</td>
<td>0.011 * _EUR</td>
</tr>
<tr>
<td>HaitianCentime</td>
<td>_cHTG</td>
<td>0.01 * _HTG</td>
</tr>
<tr>
<td>HungarianForint</td>
<td>_HUF</td>
<td>0.0032 * _EUR</td>
</tr>
<tr>
<td>HungarianFiller</td>
<td>_cHUF</td>
<td>0.01 * _HUF</td>
</tr>
<tr>
<td>IndonesianRupiah</td>
<td>_IDR</td>
<td>6.2e-05 * _EUR</td>
</tr>
<tr>
<td>IndonesianSen</td>
<td>_cIDR</td>
<td>0.01 * _IDR</td>
</tr>
<tr>
<td>IsraeliNewShekel</td>
<td>_ILS</td>
<td>0.25 * _EUR</td>
</tr>
<tr>
<td>IsraeliNewAgora</td>
<td>_cILS</td>
<td>0.01 * _ILS</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Indian Rupee</td>
<td>_INR</td>
<td>0.013 * _EUR</td>
</tr>
<tr>
<td>Indian Paisa</td>
<td>_cINR</td>
<td>0.01 * _INR</td>
</tr>
<tr>
<td>Iraqi Dinar</td>
<td>_IQD</td>
<td>0.00074 * _EUR</td>
</tr>
<tr>
<td>Iraqi Fils</td>
<td>_cIQD</td>
<td>0.001 * _IQD</td>
</tr>
<tr>
<td>Iranian Rial</td>
<td>_IRR</td>
<td>2.7e-05 * _EUR</td>
</tr>
<tr>
<td>Iranian Toman</td>
<td>_cIRR</td>
<td>10.0 * _IRR</td>
</tr>
<tr>
<td>Icelandic Krona</td>
<td>_ISK</td>
<td>0.0073 * _EUR</td>
</tr>
<tr>
<td>Jamaican Dollar</td>
<td>_JMD</td>
<td>0.007 * _EUR</td>
</tr>
<tr>
<td>Jamaican Cent</td>
<td>_cJMD</td>
<td>0.01 * _JMD</td>
</tr>
<tr>
<td>Japanese Yen</td>
<td>_JPY</td>
<td>0.008 * _EUR</td>
</tr>
<tr>
<td>Kenyan Shilling</td>
<td>_KES</td>
<td>0.0089 * _EUR</td>
</tr>
<tr>
<td>Kenyan Cent</td>
<td>_cKES</td>
<td>0.01 * _KES</td>
</tr>
<tr>
<td>Kyrgyzstani Som</td>
<td>_KGS</td>
<td>0.013 * _EUR</td>
</tr>
<tr>
<td>Kyrgyzstani Tyiyn</td>
<td>_cKGS</td>
<td>0.01 * _KGS</td>
</tr>
<tr>
<td>Cambodian Riel</td>
<td>_KHR</td>
<td>0.0022 * _EUR</td>
</tr>
<tr>
<td>North Korean Won</td>
<td>_KPW</td>
<td>0.00099 * _EUR</td>
</tr>
<tr>
<td>North Korean Chon</td>
<td>_cKPW</td>
<td>0.01 * _KPW</td>
</tr>
<tr>
<td>South Korean Won</td>
<td>_KRW</td>
<td>0.00078 * _EUR</td>
</tr>
<tr>
<td>South Korean Jeon</td>
<td>_cKRW</td>
<td>0.01 * _KRW</td>
</tr>
<tr>
<td>Kuwaiti Dinar</td>
<td>_KWD</td>
<td>2.93 * _EUR</td>
</tr>
<tr>
<td>Kuwaiti Fils</td>
<td>_cKWD</td>
<td>0.0011 * _KWD</td>
</tr>
<tr>
<td>Kazakhstani Tenge</td>
<td>_KZT</td>
<td>0.0023 * _EUR</td>
</tr>
<tr>
<td>Kazakhstani Tyiyn</td>
<td>_cKZT</td>
<td>0.01 * _KZT</td>
</tr>
<tr>
<td>Lao Kip</td>
<td>_LAK</td>
<td>0.0001 * _EUR</td>
</tr>
<tr>
<td>Lao Att</td>
<td>_cLAK</td>
<td>0.01 * _LAK</td>
</tr>
<tr>
<td>Sri Lankan Rupee</td>
<td>_LKR</td>
<td>0.005 * _EUR</td>
</tr>
<tr>
<td>Sri Lankan Cent</td>
<td>_cLKR</td>
<td>0.01 * _LKR</td>
</tr>
<tr>
<td>Liberian Dollar</td>
<td>_LRD</td>
<td>0.0055 * _EUR</td>
</tr>
<tr>
<td>Liberian Cent</td>
<td>_cLRD</td>
<td>0.01 * _LRD</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td>LibyanDinar</td>
<td>_LYD</td>
<td>0.64 * _EUR</td>
</tr>
<tr>
<td>LibyanDirham</td>
<td>_cLYD</td>
<td>0.001 * _LYD</td>
</tr>
<tr>
<td>MoroccanDirham</td>
<td>_MAD</td>
<td>0.092 * _EUR</td>
</tr>
<tr>
<td>MoroccanSantim</td>
<td>_cMAD</td>
<td>0.01 * _MAD</td>
</tr>
<tr>
<td>MoldovanLeu</td>
<td>_MDL</td>
<td>0.052 * _EUR</td>
</tr>
<tr>
<td>MoldovanBan</td>
<td>_cMDL</td>
<td>0.01 * _MDL</td>
</tr>
<tr>
<td>MalagasyAriary</td>
<td>_MGA</td>
<td>0.00025 * _EUR</td>
</tr>
<tr>
<td>MalagasyIraimbilanja</td>
<td>_cMGA</td>
<td>0.2 * _MGA</td>
</tr>
<tr>
<td>MacedonianDenar</td>
<td>_MKD</td>
<td>0.016 * _EUR</td>
</tr>
<tr>
<td>MacedonianDeni</td>
<td>_cMKD</td>
<td>0.01 * _MKD</td>
</tr>
<tr>
<td>BurmeseKyat</td>
<td>_MMK</td>
<td>0.00059 * _EUR</td>
</tr>
<tr>
<td>BurmesePya</td>
<td>_cMMK</td>
<td>0.01 * _MMK</td>
</tr>
<tr>
<td>MongolianTogrog</td>
<td>_MNT</td>
<td>0.00034 * _EUR</td>
</tr>
<tr>
<td>MongolianMongo</td>
<td>_cMNT</td>
<td>0.01 * _MNT</td>
</tr>
<tr>
<td>MauritanianOuguiya</td>
<td>_MRU</td>
<td>0.025 * _EUR</td>
</tr>
<tr>
<td>MauritanianKhoums</td>
<td>_cMRU</td>
<td>0.2 * _MRU</td>
</tr>
<tr>
<td>MauritianRupee</td>
<td>_MUR</td>
<td>0.025 * _EUR</td>
</tr>
<tr>
<td>MauritianCent</td>
<td>_cMUR</td>
<td>0.01 * _MUR</td>
</tr>
<tr>
<td>MaldivianRufiyaa</td>
<td>_MVR</td>
<td>0.058 * _EUR</td>
</tr>
<tr>
<td>MaldivianLaari</td>
<td>_cMVR</td>
<td>0.01 * _MVR</td>
</tr>
<tr>
<td>MalawianKwacha</td>
<td>_MWK</td>
<td>0.0012 * _EUR</td>
</tr>
<tr>
<td>MalawianTambala</td>
<td>_cMWK</td>
<td>0.01 * _MWK</td>
</tr>
<tr>
<td>MexicanPeso</td>
<td>_MXN</td>
<td>0.046 * _EUR</td>
</tr>
<tr>
<td>MexicanCentavo</td>
<td>_cMXN</td>
<td>0.01 * _MXN</td>
</tr>
<tr>
<td>MalaysianRinggit</td>
<td>_MYR</td>
<td>0.22 * _EUR</td>
</tr>
<tr>
<td>MalaysianSen</td>
<td>_cMYR</td>
<td>0.01 * _MYR</td>
</tr>
<tr>
<td>MozambicanMetical</td>
<td>_MZN</td>
<td>0.014 * _EUR</td>
</tr>
<tr>
<td>MozambicanCentavo</td>
<td>_cMZN</td>
<td>0.01 * _MZN</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Nigerian Naira</td>
<td>_NGN</td>
<td>0.0025 * _EUR</td>
</tr>
<tr>
<td>Nigerian Kobo</td>
<td>_cNGN</td>
<td>0.01 * _NGN</td>
</tr>
<tr>
<td>Nicaraguan Cordoba</td>
<td>_NIO</td>
<td>0.027 * _EUR</td>
</tr>
<tr>
<td>Nicaraguan Centavo</td>
<td>_cNIO</td>
<td>0.01 * _NIO</td>
</tr>
<tr>
<td>Norwegian Krone</td>
<td>_NOK</td>
<td>0.1 * _EUR</td>
</tr>
<tr>
<td>Norwegian Ore</td>
<td>_cNOK</td>
<td>0.01 * _NOK</td>
</tr>
<tr>
<td>New Zealand Dollar</td>
<td>_NZD</td>
<td>0.61 * _EUR</td>
</tr>
<tr>
<td>New Zealand Cent</td>
<td>_cNZD</td>
<td>0.01 * _NZD</td>
</tr>
<tr>
<td>Peruvian Sol</td>
<td>_PEN</td>
<td>0.27 * _EUR</td>
</tr>
<tr>
<td>Peruvian Centimo</td>
<td>_cPEN</td>
<td>0.01 * _PEN</td>
</tr>
<tr>
<td>Papua New Guinean Kina</td>
<td>_PGK</td>
<td>0.26 * _EUR</td>
</tr>
<tr>
<td>Papua New Guinean Toea</td>
<td>_cPGK</td>
<td>0.01 * _PGK</td>
</tr>
<tr>
<td>Philippine Peso</td>
<td>_PHP</td>
<td>0.017 * _EUR</td>
</tr>
<tr>
<td>Philippine Sentimo</td>
<td>_cPHP</td>
<td>0.01 * _PHP</td>
</tr>
<tr>
<td>Pakistani Rupee</td>
<td>_PKR</td>
<td>0.0064 * _EUR</td>
</tr>
<tr>
<td>Pakistani Paisa</td>
<td>_cPKR</td>
<td>0.01 * _PKR</td>
</tr>
<tr>
<td>Polish Zloty</td>
<td>_PLN</td>
<td>0.23 * _EUR</td>
</tr>
<tr>
<td>Polish Grosz</td>
<td>_cPLN</td>
<td>0.01 * _PLN</td>
</tr>
<tr>
<td>Paraguayan Guarani</td>
<td>_PYG</td>
<td>0.00015 * _EUR</td>
</tr>
<tr>
<td>Paraguayan Centimo</td>
<td>_cPYG</td>
<td>0.01 * _PYG</td>
</tr>
<tr>
<td>Qatari Riyal</td>
<td>_QAR</td>
<td>0.24 * _EUR</td>
</tr>
<tr>
<td>Qatari Dirham</td>
<td>_cQAR</td>
<td>0.01 * _QAR</td>
</tr>
<tr>
<td>Romanian Leu</td>
<td>_RON</td>
<td>0.21 * _EUR</td>
</tr>
<tr>
<td>Romanian Ban</td>
<td>_cRON</td>
<td>0.01 * _RON</td>
</tr>
<tr>
<td>Serbian Dinar</td>
<td>_RSD</td>
<td>0.0085 * _EUR</td>
</tr>
<tr>
<td>Serbian Para</td>
<td>_cRSD</td>
<td>0.01 * _RSD</td>
</tr>
<tr>
<td>Russian Ruble</td>
<td>_RUB</td>
<td>0.013 * _EUR</td>
</tr>
<tr>
<td>Russian Kopeyka</td>
<td>_cRUB</td>
<td>0.01 * _RUB</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Rwandan Franc</td>
<td>_RWF</td>
<td>0.00098 * _EUR</td>
</tr>
<tr>
<td>Rwandan Centime</td>
<td>_cRWF</td>
<td>0.01 * _RWF</td>
</tr>
<tr>
<td>Solomon Islands Dollar</td>
<td>_SBD</td>
<td>0.11 * _EUR</td>
</tr>
<tr>
<td>Solomon Islands Cent</td>
<td>_cSBD</td>
<td>0.01 * _SBD</td>
</tr>
<tr>
<td>Seychellois Rupee</td>
<td>_SCR</td>
<td>0.065 * _EUR</td>
</tr>
<tr>
<td>Seychellois Cent</td>
<td>_cSCR</td>
<td>0.01 * _SCR</td>
</tr>
<tr>
<td>Sudanese Pound</td>
<td>_SDG</td>
<td>0.019 * _EUR</td>
</tr>
<tr>
<td>Sudanese Qirsh</td>
<td>_cSDG</td>
<td>0.01 * _SDG</td>
</tr>
<tr>
<td>Swedish Krona</td>
<td>_SEK</td>
<td>0.094 * _EUR</td>
</tr>
<tr>
<td>Swedish Ore</td>
<td>_cSEK</td>
<td>0.01 * _SEK</td>
</tr>
<tr>
<td>Singapore Dollar</td>
<td>_SGD</td>
<td>0.65 * _EUR</td>
</tr>
<tr>
<td>Singapore Cent</td>
<td>_cSGD</td>
<td>0.01 * _SGD</td>
</tr>
<tr>
<td>Sierra Leonean Leone</td>
<td>_SLL</td>
<td>0.0001 * _EUR</td>
</tr>
<tr>
<td>Sierra Leonean Cent</td>
<td>_cSLL</td>
<td>0.01 * _SLL</td>
</tr>
<tr>
<td>Somaliland Shilling</td>
<td>_SQS</td>
<td>0.00013 * _EUR</td>
</tr>
<tr>
<td>Somaliland Cent</td>
<td>_cSQS</td>
<td>0.01 * _SQS</td>
</tr>
<tr>
<td>Somali Shilling</td>
<td>_SOS</td>
<td>0.0015 * _EUR</td>
</tr>
<tr>
<td>Somali Senti</td>
<td>_cSOS</td>
<td>0.01 * _SOS</td>
</tr>
<tr>
<td>Surinamese Dollar</td>
<td>_SRD</td>
<td>0.12 * _EUR</td>
</tr>
<tr>
<td>Surinamese Cent</td>
<td>_cSRD</td>
<td>0.01 * _SRD</td>
</tr>
<tr>
<td>Syrian Pound</td>
<td>_SYP</td>
<td>0.0017 * _EUR</td>
</tr>
<tr>
<td>Syrian Piastre</td>
<td>_cSYP</td>
<td>0.01 * _SYP</td>
</tr>
<tr>
<td>Thai Baht</td>
<td>_THB</td>
<td>0.028 * _EUR</td>
</tr>
<tr>
<td>Thai Satang</td>
<td>_cTHB</td>
<td>0.01 * _THB</td>
</tr>
<tr>
<td>Tajikistani Samani</td>
<td>_TJS</td>
<td>0.094 * _EUR</td>
</tr>
<tr>
<td>Tajikistani Diram</td>
<td>_cTJS</td>
<td>0.01 * _TJS</td>
</tr>
<tr>
<td>Tonganpaanga</td>
<td>_TOP</td>
<td>0.397 * _EUR</td>
</tr>
<tr>
<td>Tongan Seniti</td>
<td>_cTOP</td>
<td>0.01 * _TOP</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td>TurkishLira</td>
<td>TRY</td>
<td>0.16 * _EUR</td>
</tr>
<tr>
<td>TurkishKurus</td>
<td>cTRY</td>
<td>0.01 * _TRY</td>
</tr>
<tr>
<td>TrinidadAndTobagoDollar</td>
<td>TTD</td>
<td>0.13 * _EUR</td>
</tr>
<tr>
<td>TrinidadAndTobagoCent</td>
<td>cTTD</td>
<td>0.01 * _TTD</td>
</tr>
<tr>
<td>NewTaiwanDollar</td>
<td>TWD</td>
<td>0.029 * _EUR</td>
</tr>
<tr>
<td>NewTaiwanCent</td>
<td>cTWD</td>
<td>0.01 * _TWD</td>
</tr>
<tr>
<td>TanzanianShilling</td>
<td>TZS</td>
<td>0.00038 * _EUR</td>
</tr>
<tr>
<td>TanzanianSenti</td>
<td>cTZS</td>
<td>0.01 * _TZS</td>
</tr>
<tr>
<td>UkrainianHryvnia</td>
<td>UAH</td>
<td>0.00038 * _EUR</td>
</tr>
<tr>
<td>UkrainianKopiyka</td>
<td>cUAH</td>
<td>0.01 * _UAH</td>
</tr>
<tr>
<td>UgandanShilling</td>
<td>UGX</td>
<td>0.00024 * _EUR</td>
</tr>
<tr>
<td>UgandanCent</td>
<td>cUGX</td>
<td>0.01 * _UGX</td>
</tr>
<tr>
<td>US Dollar</td>
<td>USD</td>
<td>0.89 * _EUR</td>
</tr>
<tr>
<td>US Cent</td>
<td>cUSD</td>
<td>0.01 * _USD</td>
</tr>
<tr>
<td>UruguayanPeso</td>
<td>UYU</td>
<td>0.027 * _EUR</td>
</tr>
<tr>
<td>UruguayanCentesimo</td>
<td>cUYU</td>
<td>0.01 * _UYU</td>
</tr>
<tr>
<td>UzbekistaniSom</td>
<td>UZS</td>
<td>0.00011 * _EUR</td>
</tr>
<tr>
<td>UzbekistaniCentesimo</td>
<td>cUZS</td>
<td>0.01 * _UZS</td>
</tr>
<tr>
<td>VenezuelanBolivarSoberano</td>
<td>VES</td>
<td>0.0003 * _EUR</td>
</tr>
<tr>
<td>VenezuelanCentesimoSoberano</td>
<td>cVES</td>
<td>0.01 * _VES</td>
</tr>
<tr>
<td>VietnameseDong</td>
<td>VND</td>
<td>3.8e-05 * _EUR</td>
</tr>
<tr>
<td>VietnameseXu</td>
<td>cVND</td>
<td>0.01 * _VND</td>
</tr>
<tr>
<td>SamoanTala</td>
<td>WST</td>
<td>0.34 * _EUR</td>
</tr>
<tr>
<td>SamoanSene</td>
<td>cWST</td>
<td>0.01 * _WST</td>
</tr>
<tr>
<td>YemeniRial</td>
<td>YER</td>
<td>0.0036 * _EUR</td>
</tr>
<tr>
<td>YemeniDinar</td>
<td>cYER</td>
<td>0.01 * _YER</td>
</tr>
<tr>
<td>SouthAfricanRand</td>
<td>ZAR</td>
<td>0.062 * _EUR</td>
</tr>
<tr>
<td>SouthAfricanCent</td>
<td>cZAR</td>
<td>0.01 * _ZAR</td>
</tr>
<tr>
<td>ZambianKwacha</td>
<td>ZMW</td>
<td>0.074 * _EUR</td>
</tr>
<tr>
<td>ZambianNgwee</td>
<td>cZMW</td>
<td>0.01 * _ZMW</td>
</tr>
</tbody>
</table>
### 4.10.1 Pegged International Currencies

International currency which are pegged to other currencies.

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>United Arab Emirates Dirham</td>
<td>_AED</td>
<td>(1/3.6725) * _USD</td>
</tr>
<tr>
<td>United Arab Emirates Fils</td>
<td>_cAED</td>
<td>0.01 * _AED</td>
</tr>
<tr>
<td>Netherlands Antillean Guilder</td>
<td>_ANG</td>
<td>(1/1.79) * _USD</td>
</tr>
<tr>
<td>Netherlands Antillean Cent</td>
<td>_cANG</td>
<td>0.01 * _ANG</td>
</tr>
<tr>
<td>Aruban Florin</td>
<td>_AWG</td>
<td>(1/1.79) * _USD</td>
</tr>
<tr>
<td>Aruban Cent</td>
<td>_cAWG</td>
<td>0.01 * _AWG</td>
</tr>
<tr>
<td>Barbadian Dollar</td>
<td>_BBD</td>
<td>0.5 * _USD</td>
</tr>
<tr>
<td>Barbadian Cent</td>
<td>_cBBD</td>
<td>0.01 * _BBD</td>
</tr>
<tr>
<td>Bulgarian Lev</td>
<td>_BGN</td>
<td>0.51129 * _EUR</td>
</tr>
<tr>
<td>Bulgarian Stotinka</td>
<td>_cBGN</td>
<td>0.01 * _BGN</td>
</tr>
<tr>
<td>Bahraini Dinar</td>
<td>_BHD</td>
<td>(1/0.376) * _USD</td>
</tr>
<tr>
<td>Bahraini Fils</td>
<td>_cBHD</td>
<td>0.001 * _BHD</td>
</tr>
<tr>
<td>Bermudian Dollar</td>
<td>_BMD</td>
<td>1 * _USD</td>
</tr>
<tr>
<td>Bermudian Cent</td>
<td>_cBMD</td>
<td>0.01 * _BMD</td>
</tr>
<tr>
<td>Brunei Dollar</td>
<td>_BND</td>
<td>1 * _SGD</td>
</tr>
<tr>
<td>Brunei Sen</td>
<td>_cBND</td>
<td>0.01 * _BND</td>
</tr>
<tr>
<td>Bahamian Dollar</td>
<td>_BSD</td>
<td>1 * _USD</td>
</tr>
<tr>
<td>Bahamian Cent</td>
<td>_cBSD</td>
<td>0.01 * _BSD</td>
</tr>
<tr>
<td>Bhutanese Ngultrum</td>
<td>_BTN</td>
<td>1 * _INR</td>
</tr>
<tr>
<td>Bhutanese Chhertum</td>
<td>_cBTN</td>
<td>0.01 * _BTN</td>
</tr>
<tr>
<td>Belize Dollar</td>
<td>_BZD</td>
<td>0.5 * _USD</td>
</tr>
<tr>
<td>Belize Cent</td>
<td>_cBZD</td>
<td>0.01 * _BZD</td>
</tr>
<tr>
<td>Cubano Convertible Peso</td>
<td>_CUC</td>
<td>1 * _USD</td>
</tr>
<tr>
<td>Cubano Convertible Centavo</td>
<td>_cCUC</td>
<td>0.01 * _CUC</td>
</tr>
<tr>
<td>Cuban Peso</td>
<td>_CUP</td>
<td>(1/24) * _CUC</td>
</tr>
<tr>
<td>Cuban Centavo</td>
<td>_cCUP</td>
<td>0.01 * _CUP</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>CapeVerdeanEscudo</td>
<td>_CVE</td>
<td>(1/110.265) * _EUR</td>
</tr>
<tr>
<td>CapeVerdeanCentavo</td>
<td>_cCVE</td>
<td>0.01 * _CVE</td>
</tr>
<tr>
<td>DjiboutianFranc</td>
<td>_DJF</td>
<td>(1/177.721) * _USD</td>
</tr>
<tr>
<td>DjiboutianCentime</td>
<td>_cDJF</td>
<td>0.01 * _DJF</td>
</tr>
<tr>
<td>EritreanNakfa</td>
<td>_ERN</td>
<td>(1/15) * _USD</td>
</tr>
<tr>
<td>EritreanCent</td>
<td>_cERN</td>
<td>0.01 * _ERN</td>
</tr>
<tr>
<td>FalklandIslandsPound</td>
<td>_FKP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>FalklandIslandsPenny</td>
<td>_cFKP</td>
<td>0.01 * _FKP</td>
</tr>
<tr>
<td>GuernseyPound</td>
<td>_GGP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>GuernseyPenny</td>
<td>_cGGP</td>
<td>0.01 * _GGP</td>
</tr>
<tr>
<td>GibraltarPound</td>
<td>_GIP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>GibraltarPenny</td>
<td>_cGIP</td>
<td>0.01 * _GIP</td>
</tr>
<tr>
<td>ManxPound</td>
<td>_IMP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>ManxPenny</td>
<td>_cIMP</td>
<td>0.01 * _IMP</td>
</tr>
<tr>
<td>JerseyPound</td>
<td>_JEP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>JerseyPenny</td>
<td>_cJEP</td>
<td>0.01 * _JEP</td>
</tr>
<tr>
<td>JordanianDinar</td>
<td>_JOD</td>
<td>(1/0.708) * _USD</td>
</tr>
<tr>
<td>JordanianFils</td>
<td>_cJOD</td>
<td>0.001 * _JOD</td>
</tr>
<tr>
<td>KiribatiDollar</td>
<td>_KID</td>
<td>1 * _AUD</td>
</tr>
<tr>
<td>KiribatiCent</td>
<td>_cKID</td>
<td>0.01 * _KID</td>
</tr>
<tr>
<td>Comorianfranc</td>
<td>_KMF</td>
<td>(1/491.96775) * _EUR</td>
</tr>
<tr>
<td>ComorianCentime</td>
<td>_cKMF</td>
<td>0.01 * _KMF</td>
</tr>
<tr>
<td>CaymanIslandsDollar</td>
<td>_KYD</td>
<td>1.2 * _USD</td>
</tr>
<tr>
<td>CaymanIslandsCent</td>
<td>_cKYD</td>
<td>0.01 * _KYD</td>
</tr>
<tr>
<td>LebanesePound</td>
<td>_LBP</td>
<td>(1/1507.5) * _USD</td>
</tr>
<tr>
<td>LebaneseQeresh</td>
<td>_cLBP</td>
<td>0.01 * _LBP</td>
</tr>
<tr>
<td>MacanesePataca</td>
<td>_MOP</td>
<td>(1/1.03) * _HKD</td>
</tr>
<tr>
<td>MacaneseAvo</td>
<td>_cMOP</td>
<td>0.01 * _MOP</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Namibian Dollar</td>
<td>_NAD</td>
<td>1 * _ZAR</td>
</tr>
<tr>
<td>Namibian Cent</td>
<td>_cNAD</td>
<td>0.01 * _NAD</td>
</tr>
<tr>
<td>Nepalese Rupee</td>
<td>_NPR</td>
<td>(1/1.6) * _INR</td>
</tr>
<tr>
<td>Nepalese Paisa</td>
<td>_cNPR</td>
<td>0.01 * _NPR</td>
</tr>
<tr>
<td>Omani Rial</td>
<td>_OMR</td>
<td>(1/2.6008) * _USD</td>
</tr>
<tr>
<td>Omani Baisa</td>
<td>_cOMR</td>
<td>0.001 * _OMR</td>
</tr>
<tr>
<td>Panamanian Balboa</td>
<td>_PAB</td>
<td>1 * _USD</td>
</tr>
<tr>
<td>Panamanian Centesimo</td>
<td>_cPAB</td>
<td>0.01 * _PAB</td>
</tr>
<tr>
<td>Transnistrian Ruble</td>
<td>_PRB</td>
<td>(1/16.1) * _USD</td>
</tr>
<tr>
<td>Transnistrian Kopeck</td>
<td>_cPRB</td>
<td>0.01 * _PRB</td>
</tr>
<tr>
<td>Saudi Riyal</td>
<td>_SAR</td>
<td>(1/3.75) * _USD</td>
</tr>
<tr>
<td>Saudi Halalah</td>
<td>_cSAR</td>
<td>0.01 * _SAR</td>
</tr>
<tr>
<td>Saint Helena Pound</td>
<td>_SHP</td>
<td>1 * _GBP</td>
</tr>
<tr>
<td>Saint Helena Penny</td>
<td>_cSHP</td>
<td>0.01 * _SHP</td>
</tr>
<tr>
<td>South Sudanese Pound</td>
<td>_SSP</td>
<td>1 * _SDG</td>
</tr>
<tr>
<td>South Sudanese Piaster</td>
<td>_cSSP</td>
<td>0.01 * _SSP</td>
</tr>
<tr>
<td>Sao Tome And Principe Dobra</td>
<td>_STN</td>
<td>(1/24.5) * _EUR</td>
</tr>
<tr>
<td>Sao Tome And Principe Centimo</td>
<td>_cSTN</td>
<td>0.01 * _STN</td>
</tr>
<tr>
<td>Swazi Lilangeni</td>
<td>_SZL</td>
<td>1 * _ZAR</td>
</tr>
<tr>
<td>Swazi Cent</td>
<td>_cSZL</td>
<td>0.01 * _SZL</td>
</tr>
<tr>
<td>Turkmenistan Manat</td>
<td>_TMT</td>
<td>(1/3.5) * _USD</td>
</tr>
<tr>
<td>Turkmenistan Tenge</td>
<td>_cTMT</td>
<td>0.01 * _TMT</td>
</tr>
<tr>
<td>Tuvaluan Dollar</td>
<td>_TVD</td>
<td>1 * _AUD</td>
</tr>
<tr>
<td>Tuvaluan Cent</td>
<td>_cTVD</td>
<td>0.01 * _TVD</td>
</tr>
<tr>
<td>Central African CFA Franc</td>
<td>_XAF</td>
<td>(1/655.957) * _EUR</td>
</tr>
<tr>
<td>Central African CFA Centime</td>
<td>_cXAF</td>
<td>0.01 * _XAF</td>
</tr>
<tr>
<td>Eastern Caribbean Dollar</td>
<td>_XCD</td>
<td>(1/2.7) * _USD</td>
</tr>
<tr>
<td>Eastern Caribbean Cent</td>
<td>_cXCD</td>
<td>0.01 * _XCD</td>
</tr>
<tr>
<td>Name</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td>West African CFA Franc</td>
<td>_XOF</td>
<td>(1/655.957) * _USD</td>
</tr>
<tr>
<td>West African CFA Centime</td>
<td>_cXOF</td>
<td>0.01 * _XOF</td>
</tr>
<tr>
<td>CFP Franc</td>
<td>_XPF</td>
<td>(1000/8.38) * _EUR</td>
</tr>
<tr>
<td>CFP Centime</td>
<td>_cXPF</td>
<td>0.01 * _XPF</td>
</tr>
<tr>
<td>Zimbabwean Bonds</td>
<td>_ZWL</td>
<td>1 * _USD</td>
</tr>
<tr>
<td>Zimbabwean Cent</td>
<td>_cZWL</td>
<td>0.01 * _ZWL</td>
</tr>
</tbody>
</table>
5   Lua Documentation

In this section, the following shortcuts will be used.

```lua
local D = physical.Dimension
local U = physical.Unit
local N = physical.Number
local Q = physical.Quantity
```

The term **number** refers to a lua integer or a lua float number. By **string** a lua string is meant and by **bool** a lua boolean.

5.1   physical.Quantity

The quantity class is the main part of the library. Each physical Quantity and all units are represented by an instance of this class.

**Q.new(q=nil)**

Copy Constructor

```
q : Q, number, object, nil
```

returns : Q

As an argument it takes **Q, number, object** or **nil**. If an instance of **Q** is given, a copy is made and returned. If a **number** or an instance **object** of another class is given, the function creates a dimensionless quantity with the **number** or the instance as a value. In the case **nil** is given, a dimensionless quantity with value 1 is returned.

```lua
print( Q() )
1
print( Q(42) )
42
print( Q(73*_m) )
73 * _m
```

**Q.defineBase(symbol,name,dimension)**

This function is used to declare base quantities from which all other quantities are derived from.

```javascript
symbol : string
The symbol of the base quantity.
```
name : string
    The name of the base quantity.

dimension : D
    An instance of the D class, which represents the dimension of the quantity.

returns : Q
    The created Q instance.

The function creates a global variable of the created base quantity. The name consist of an underscore concatenated with the symbol argument, i.e. the symbol m becomes the global variable _m.

The name is used for example in the siunitx conversion function, e.g meter will be converted to \meter.

Each quantity has a dimension associated with it. The argument dimension allows any dimension to be associated to base quantities.

```
Q.defineBase("m", "meter", L)
Q.defineBase("kg", "kilogram", M)
```

**Q.define(symbol, name, q)**

Creates a new derived quantity from an expression of other quantities. Affine quantities like the absolute temperature in celsius are not supported.

symbol : string
    Symbol of the base quantity

name : string, nil
    The Name of the derived quantity.

q : physical.Quantity
    The definition of the derived quantity.

returns : Q
    The created quantity.

The function creates a global variable of the created base quantity. The name consist of an underscore concatenated with the symbol argument, i.e. the symbol N becomes the global variable _N.

The name is used for example in the siunitx conversion function, e.g newton will be converted to \newton.

```
Q.define("L", "liter", _dm^3)
Q.define("Pa", "pascal", _N/_m^2)
Q.define("C", "coulomb", _A*_s)
Q.define("degC", "celsius", _K)
```
Q.definePrefix(symbol, name, factor)
Definitions a new prefix.

symbol : string
Symbol of the base quantity

name : string
Name of the base quantity

factor : number
The factor which corresponds to the prefix

Q.definePrefix("c", "centi", 1e-2)
Q.definePrefix("a", "atto", 1e-18)

Q.addPrefix(prefixes, units)
Create several units with prefixes from a given unit.

prefixes : string
A list of unit symbols.

units : Q
A list of quantities.

Q.addPrefix({'n','u','m','k','M','G'},{_m,_s,_A})

Q.isclose(self, q, r)
Checks if this quantity is close to another one. The argument r is the maximum relative deviation. The function returns true if the following condition is fullfilled

\[
\frac{|self - q|}{\min(|self|, |q|)} \leq r
\]

self : Q, N, number
q : Q, N, number
r : number
maximum relative deviation of self and q
returns : bool
true if q is close to self, otherwise false
s_1 = 1.9 * _m
s_2 = 2.0 * _m
print( s_1:isclose(s_2,0.1) )
true
print( s_1:isclose(s_2,0.01) )
false

Q.to(self,q=nil)

Converts the quantity self to the unit of the quantity q. If no q is given, the quantity self is converted to base units.

self : Q
q : Q, nil

s = 1.9 * _km
print( s.to(_m) )
1900.0 * _m
T = 10 * _degC
print( T.to(_K) )
10.0 * _K
print( T.to() )
10 * _K

Q.tosiunitx(self,param,mode=Q.siunitx_SI)

Converts the quantity into a siunitx string.

self : Q
param : string
mode : number

If mode is equal Q.SIUNITX_SI, which is the default, the quantity is converted to an \SI{}{} macro. If mode is Q.SIUNITX_num, the quantity is converted to \num{} and if it is Q.SIUNITX_si the macro \si{} is printed.

s = 1.9 * _km
print( s.tosiunitx() )
\SI{1.9}{\kilo\meter}
print( s.tosiunitx(nil,Q.SIUNITX_num) )
\SI{1.9}{\kilo\meter}
print( s.tosiunitx(nil,Q.SIUNITX_si) )
\SI{1.9}{\kilo\meter}
\textbf{Q.min(q1, q2, ...)}

Returns the smallest quantity of the given ones. The function returns \texttt{q1} if the Quantities are equal.

\texttt{q1 : Q, N, number}
\texttt{q2 : Q, N, number}
...
\texttt{qN : Q, N, number}

\texttt{returns : Q}
the smallest quantity of \texttt{q1, ..., qN}

\begin{verbatim}
  s_1 = 15 * _m
  s_2 = 5 * _m
  print(s_1:min(s_2))
  5 * _m
\end{verbatim}

\textbf{Q.max(q1, q2, ...)}

Returns the biggest quantity of several given ones. The function returns \texttt{q1} if the Quantities are equal.

\texttt{q1 : Q, N, number}
\texttt{q2 : Q, N, number}
...
\texttt{qN : Q, N, number}

\texttt{returns : Q}
the biggest quantity of \texttt{q1, ..., qN}

\begin{verbatim}
  s_1 = 15 * _m
  s_2 = 5 * _m
  print(s_1:max(s_2))
  15 * _m
\end{verbatim}

\textbf{Q.abs(q)}

Returns the absolute value of the given quantity \texttt{q}.

\texttt{q : Q, N, number}

\texttt{returns : Q}
the absolute value of \texttt{q}

\begin{verbatim}
  s_1 = 15 * _m
  s_2 = 5 * _m
  print(s_1:abs(s_2))
  15 * _m
\end{verbatim}
\[ U = -5 \cdot _V \]
\[ \text{print}(U) \]
\[ -5 \cdot _V \]
\[ \text{print}(U; \text{abs}()) \]
\[ 5 \cdot _V \]

\[ Q.\text{sqrt}(q) \]

Returns the square root of the given quantity.

\[ q : Q, N, \text{number} \]
\hspace{10pt} \text{dimensionless argument} \\
\text{return} \hspace{10pt} Q \\
\hspace{10pt} \text{the square root of } q \\

\[ A = 25 \cdot _m^2 \]
\[ s = A;\text{sqrt}() \]
\[ \text{print}(s) \]
\[ 5.0 \cdot _m \]

\[ Q.\text{log}(q, \text{base}=\text{nil}) \]

Returns the logarithm of a given quantity to the given base. If no base is given, the natural logarithm is returned.

\[ q : Q, N, \text{number} \]
\hspace{10pt} \text{dimensionless argument} \\
\text{base} \hspace{10pt} Q, N, \text{number}, \text{nil} \\
\hspace{10pt} \text{dimensionless argument} \\
\text{return} \hspace{10pt} Q \\
\hspace{10pt} \text{logarithm of } q \text{ to the } \text{base} \\

\[ I = 1 \cdot _W/_m^2 \]
\[ I_0 = 1e-12 \cdot _W/_m^2 \]
\[ \text{print}(10 \cdot (I/I_0);\text{log}(10) \cdot _\text{dB}) \]
\[ 120.0 \cdot _\text{dB} \]

\[ Q.\text{exp}(q) \]

Returns the value of the natural exponential function of the given quantity.

\[ q : Q, N, \text{number} \]
\hspace{10pt} \text{dimensionless argument}
returns: Q
natural exponential of q

\[
x = 2 \cdot 1_
\]
print( x:exp() )
7.3890560989307

\qquad Q.\sin(q) \\
\quad Returns the value of the sinus function of the given quantity.
\qquad q: Q, N, number \\
\quad dimensionless argument
\qquad returns: Q \\
\quad sine of q

\[
alpha = 30 \cdot \_\text{deg}
\]
print( alpha:sin() )
0.5

\qquad Q.\cos(q) \\
\quad Returns the value of the cosinus function of the given quantity. The quantity has to be dimensionless.
\qquad q: Q, N, number \\
\quad dimensionless argument
\qquad returns: Q \\
\quad cosine of q

\[
alpha = 60 \cdot \_\text{deg}
\]
print( alpha:cos() )
0.5

\qquad Q.\tan(q) \\
\quad Returns the value of the tangent function of the given quantity. The quantity has to be dimensionless.
\qquad q: Q, N, number \\
\quad dimensionless argument
\qquad returns: Q \\
\quad tangent of q

41
alpha = 45 * _deg
print( alpha:tan() )
1.0

Q.asin(q)

Returns the value of the arcus sinus function of the given quantity. The quantity has to be dimensionless.

q : Q, N, number
dimensionless argument
returns : Q
inverse sine of q

x = 0.5 * _1
print( x:asin():to(_deg) )
30.0 * _deg

Q.acos(q)

Returns the value of the arcus cosinus function of the given quantity. The quantity has to be dimensionless.

q : Q, N, number
dimensionless argument
returns : Q
inverse cosine of q

x = 0.5 * _1
print( x:acos():to(_deg) )
60.0 * _deg

Q.atan(q)

Returns the value of the arcus tangent function of the given quantity. The quantity has to be dimensionless.

q : Q, N, number
dimensionless argument
returns : Q
inverse tangent of q

x = 1 * _1
print( x:atan():to(_deg) )
45.0 * _deg
Q.sinh(q)
Returns the value of the hyperbolic sine function of the given quantity. 
The quantity has to be dimensionless. Since Lua doesn’t implement 
the hyperbolic functions, the following formula is used 
\[
\sinh(x) = 0.5 \cdot e^x - 0.5/e^x 
\]
q : Q, N, number 
dimensionless argument 
returns : Q 
hyperbolic sine of q 

```lua
x = 1 * _1
print( x:sinh() )
1.1752011936438
```

Q.cosh(q)
Returns the value of the hyperbolic cosine function of the given quan-
tity. The quantity has to be dimensionless. Since Lua doesn’t imple-
ment the hyperbolic functions, the following formula is used 
\[
cosh(x) = 0.5 \cdot e^x + 0.5/e^x 
\]
q : Q, N, number 
dimensionless argument 
returns : Q 
hyperbolic cosine of q 

```lua
x = 1 * _1
print( x:cosh() )
1.5430806348152
```

Q.tanh(q)
Returns the value of the hyperbolic tangent function of the given quan-
tity. The quantity has to be dimensionless. Since Lua doesn’t imple-
ment the hyperbolic functions, the following formula is used 
\[
tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} 
\]
q : Q, N, number 
dimensionless argument 
returns : Q 
hyperbolic tangent of q 

43
\[ x = 1 * \_1 \]
\[ \text{print}(x: \text{tanh}()) \]
\[ 0.76159415595576 \]

\textbf{Q.asinh(q)}

Returns the value of the inverse hyperbolic sine function of the given quantity. The quantity has to be dimensionless. Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{asinh}(x) = \ln \left( x + \sqrt{x^2 + 1} \right) . \]

\( q \): Q, N, number
- dimensionless argument

\( \text{returns: Q} \)
- inverse hyperbolic sine of \( q \)

\[ x = 1 * \_1 \]
\[ \text{print}(x: \text{asinh}()) \]
\[ 0.88137358701954 \]

\textbf{Q.acosh(q)}

Returns the value of the inverse hyperbolic cosine function of the given quantity. The quantity has to be dimensionless. Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{acosh}(x) = \ln \left( x + \sqrt{x^2 - 1} \right) , x > 1 . \]

\( q \): Q, N, number
- dimensionless argument bigger or equal to one

\( \text{returns: Q} \)
- inverse hyperbolic cosine of \( q \)

\[ x = 2 * \_1 \]
\[ \text{print}(x: \text{acosh}()) \]
\[ 1.3169578969248 \]

\textbf{Q.atanh(q)}

Returns the value of the inverse hyperbolic tangent function of the given quantity. The quantity has to be dimensionless. Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{atanh}(x) = \ln \left( \frac{1 + x}{1 - x} \right) , -1 < x < 1 . \]
q : Q, N, number
    dimensionless argument with magnitude smaller than one
returns : Q
    inverse hyperbolic tangent of q

```python
x = 0.5 * _1
print( x:atanh() )
0.54930614433405
```
5.2 physical.Dimension

All physical quantities do have a physical dimension. For example the quantity *Area* has the dimension $L^2$ (length to the power of two). In the SI-System there are seven base dimensions, from which all other dimensions are derived. Each dimension is represented by an n-tuple, where n is the number of base dimensions. Each physical quantity has an associated dimension object. It is used to check equality and if addition or substraction is allowed.

**D.new(d)**

Constructor of the *Dimension* class.

`d : Dimension or string, nil`

The name or symbol of the dimension.

returns : D

The created D instance

If `d` is a string, a copy of the perviously defined dimension is made. If `d` is a dimension, a copy of it is made. If no argument ist given, a dimension zero is created.

**Example**

```plaintext
V_1 = D("Velocity")
L = D("L")
V_2 = D(L/T)
```

**D.defineBase(symbol, name)**

Defines a base dimension.

`symbol : string`

`name : string`

returns : D

The created D instance

**Example**

```plaintext
V_1 = D("Velocity")
L = D("L")
V_2 = D(L/T)
```
5.3 physical.Unit

The task of this class is keeping track of the unit term. The unit term is a fraction of units. The units in the enumerator and denominator can have an exponent.

**Unit.new(u=nil)**

Copy Constructor. It copies a given unit object. If nothing is given, an empty unit is created.

- **u**: Unit
  - The unit object which will be copied.

returns : Unit
  - The created Unit object

**Unit.new(symbol, name, prefixsymbol=nil, prefixname=nil)**

Constructor. A new Unit object with symbol is created. The prefixsymbol and prefixname are optional.

- **symbol**: String
  - The symbol of the unit.
- **name**: String
  - The name of the unit.
- **prefixsymbol**: String
  - The optional symbol of the prefix.
- **prefixname**: String
  - The optional name of the prefix.

returns : Unit
  - The created Unit object

**Unit.tosiunitx(self)**

The unit term will be compiled into a string, which the LaTeX package siunitx can understand.

returns : String
  - The siunitx representation of the unit term.
5.4 physical.Number

It does arithmetics with gaussian error propagation. A number instance has a mean value called \( x \) and an uncertainty value called \( dx \).

\[ \text{N.new(n=nil)} \]

This is the copy Constructor. It copies a given number object. If \( n \) is nil, an instance representing number zero with uncertainty zero is created.

\[ n : \text{Number} \]

The number object to be copied.

\[ \text{returns : Number} \]

The created \text{Number} instance.

\[
\begin{align*}
n &= \text{N}(56, 0.012) \\
m &= \text{N}(n) \\
\text{print}(m) & \quad (56.000 +/- 0.012)
\end{align*}
\]

\[ \text{N.new(x, dx=nil)} \]

This constructor creates a new instance of \text{N} with mean value \( x \) and uncertainty \( dx \). If \( dx \) is not given, the uncertainty is zero.

\[ x : \text{number} \]

mean value

\[ dx : \text{number, nil} \]

uncertainty value

\[ \text{returns : N} \]

The created \text{N} instance.

\[
\begin{align*}
n &= \text{N}(56, 0.012) \\
\text{print}(n) & \quad (56.000 +/- 0.012)
\end{align*}
\]

\[ \text{N.new(str)} \]

This constructor creates a new instance of \text{N} from a string. It can parse strings of the form "3.4", "3.4e-3", "5.4e-3 +/- 2.4e-6" and "5.45(7)e-23".

\[ \text{str : string} \]

\[ \text{returns : N} \]
\begin{verbatim}
  n_1 = N("12.3e-3")
  print(n_1)
  (0.01230 +/- 0.00005)

  n_2 = N("12 +/- 0.1")
  print(n_2)
  (12.00 +/- 0.10)

  n_3 = N("12.0(1)")
  print(n_3)
  (12.00 +/- 0.10)

  n_4 = N("15.0(12)")
  print(n_4)
  (15.0 +/- 1.2)
\end{verbatim}

\textbf{N.mean(n)}

Returns the mean value of \(n\).

\textbf{Parameters / Return}

\texttt{returns : number}

\begin{verbatim}
  n = N(1.25,0.0023)
  print( n:mean() )
  1.25
\end{verbatim}

\textbf{N.uncertainty(n)}

Returns the uncertainty value of \(n\).

\(n : N\)

\texttt{returns : number}

\begin{verbatim}
  n = N(1.25,0.0023)
  print( n:uncertainty() )
  0.0023
\end{verbatim}

\textbf{N.abs(n)}

Returns the absolute value of \(n\).

\(n : N\)

\texttt{returns : N}

49
The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \Delta x .$$

```
N = N(-10,1)
print(N.abs())
(10.0 +/- 1.0)
```

**N.sqrt(n)**

Returns the square root of $n$.

```
n = N
returns = N
```

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \frac{1}{2\sqrt{x}} \cdot \Delta x .$$

```
n = N(25,1)
print(n:sqrt())
(5.00 +/- 0.10)
```

**N.log(n,base=nil)**

Returns the logarithm of a given number $n$ to the given base $base$. If no base is given, the natural logarithm of $n$ is returned.

```
n = N
base = number, nil
returns = N
```

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \frac{1}{|x \cdot \log(b)|} \cdot \Delta x .$$

```
n = N(25,1)
print(n:log())
(3.22 +/- 0.04)
```
N.exp(n)
Returns the value of the natural exponential function of the given number.

q : N
returns : N

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = e^x \cdot \Delta x$$

```
n = N(25,1)
print( n:sqrt() )
(5.00 +/- 0.10)
```

N.sin(n)
Returns the value of the sine function of the given number.

n : N
returns : N

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = |\cos(x)| \cdot \Delta x$$

```
n = N(3,0.1)
print( n:sin() )
(0.14 +/- 0.10)
```

N.cos(n)
Returns the value of the cosine function of the given number.

n : N
returns : N

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = |\sin(x)| \cdot \Delta x$$

```
n = N(0.5,0.01)
print( n:cos() )
(0.878 +/- 0.005)
```
\textbf{N.tan(n)}

Returns the value of the tangent function of the given number.

\begin{verbatim}
n : N
returns : N
\end{verbatim}

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \frac{1}{\cos^2(x)} \cdot \Delta x .$$

\begin{verbatim}
n = N(1.5, 0.01)
print( n.tan() )
(14.1 +/- 2.0)
\end{verbatim}

\textbf{N.asin(n)}

Returns the value of the inverse sine function of the given number.

\begin{verbatim}
n : N
returns : N
\end{verbatim}

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \frac{1}{\sqrt{1 - x^2}} \cdot \Delta x .$$

\begin{verbatim}
n = N(0.99, 0.1)
print( n.asin() )
(1.4 +/- 0.7)
\end{verbatim}

\textbf{N.acos(n)}

Returns the value of the inverse cosine function of the given number.

\begin{verbatim}
n : N
returns : N
\end{verbatim}

The uncertainty $\Delta y$ is calculated by the following expression

$$\Delta y = \frac{1}{\sqrt{1 - x^2}} \cdot \Delta x .$$

\begin{verbatim}
n = N(0.99, 0.1)
print( n.acos() )
(0.1 +/- 0.7)
\end{verbatim}

52
**N.atan(n)**

Returns the value of the inverse tangent function of the given number.

\[ n : N \]

\[ \text{returns : } N \]

The uncertainty \( \Delta y \) is calculated by the following expression

\[ \Delta y = \frac{1}{\sqrt{1+x^2}} \cdot \Delta x. \]

```lua
n = N(1,0.1)
print( n:atan() )
(0.79 +/- 0.05)
```

**N.sinh(q)**

Returns the value of the hyperbolic sine function of the given number.

\[ n : N \]

\[ \text{returns : } N \]

Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \sinh(x) = 0.5 \cdot e^x - 0.5/e^x. \]

The uncertainty \( \Delta y \) is calculated by the following expression

\[ \Delta y = (0.5 \cdot e^x + 0.5/e^x) \cdot \Delta x. \]

```lua
n = N(1,0.1)
print( n:sinh() )
(1.18 +/- 0.15)
```

**N.cosh(q)**

Returns the value of the hyperbolic cosine function of the given number.

\[ n : N \]

\[ \text{returns : } N \]
Since Lua doesn’t implement the hyperbolic functions, the following formula is used
\[
\cosh(x) = 0.5 \cdot e^x + 0.5/e^x .
\]
The uncertainty $\Delta y$ is calculated by the following expression
\[
\Delta y = (0.5 \cdot e^x - 0.5/e^x) \cdot \Delta x .
\]

```lua
n = N(1,0.1)
print(n:cosh())
(1.54 +/- 0.12)
```

**N.tanh(q)**

Returns the value of the hyperbolic tangent function of the given number.

```lua
n : N
returns : N
```

Since Lua doesn’t implement the hyperbolic functions, the following formula is used
\[
\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} .
\]
The uncertainty $\Delta y$ is calculated by the following expression
\[
\Delta y = \frac{1}{(0.5 \cdot e^x + 0.5/e^x)^2} \cdot \Delta x .
\]

```lua
n = N(1,0.1)
print(n:tanh())
(0.76 +/- 0.04)
```

**Q.asinh(q)**

Returns the value of the inverse hyperbolic sine function of the given number.

```lua
n : N
returns : N
```
Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{asinh}(x) = \ln(x + \sqrt{x^2 + 1}) \].

The uncertainty \( \Delta y \) is calculated by the following expression

\[ \Delta y = \frac{1}{\sqrt{x^2 + 1}} \cdot \Delta x . \]

```lua
n = N(1,0.1)
print( n:asinh() )
(0.88 +/- 0.07)
```

**Q.acosh(q)**

Returns the value of the inverse hyperbolic cosine function of the given number.

```lua
n : N
returns : N
```

Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{acosh}(x) = \ln(x + \sqrt{x^2 - 1}) \quad , x > 1 \].

The uncertainty \( \Delta y \) is calculated by the following expression

\[ \Delta y = \frac{1}{\sqrt{x^2 - 1}} \cdot \Delta x . \]

```lua
n = N(1,0.1)
print( n:acosh() )
(0 +/- inf)
```

**Q.atanh(q)**

Returns the value of the inverse hyperbolic tangent function of the given number.

```lua
n : N
returns : N
```
Since Lua doesn’t implement the hyperbolic functions, the following formula is used

\[ \text{atanh}(x) = \ln\left(\frac{1 + x}{1 - x}\right), \quad -1 < x < 1. \]

The uncertainty \( \Delta y \) is calculated by the following expression

\[ \Delta y = \frac{1}{|x^2 - 1|} \cdot \Delta x. \]

```lua
n = N(1,0.1)
print( n:atanh() )
(inf +/- inf)
```
6 Change History

V1.0.3 (2020/09/09)  Minor release
   Changed foldername physical to src. Changed Classvariables Q.siunitx_SI,
   Q.siunitx_num and Q.siunitx_si to uppercase Q.SIUNITX_SI, Q.SIUNITX_num
   and Q.SIUNITX_si.

V1.0.2 (2020/09/07)  Minor release
   Path issues resolved. Docs corrected.

V1.0.1 (2020/09/05)  Minor release
   Files renamed.

V1.0.1 (2020/09/03)  First official release.
References


Index of Units

acre _ac, 20
amperehour _Ah, 19
amperesecond _As, 19
angstrom _angstrom, 18
arcminute _arcmin, 17
arcsecond _arcsec, 17
are _are, 18
astronomicalunit _au, 18
atomicmassunit , 15
avogadronumber , 15
bar _bar, 18
barn _barn, 18
becquerel _Bq, 16
bitpersecond _bps, 19
bohrmagneton , 15
boltzmannconstant , 15
byte _B, 19
cable _cbl, 20
celsius _degC, 16
chain _ch, 20
coulomb _C, 16
cup _cup, 20
curie _Ci, 19
day _d, 17
decibel _dB, 19
degree _deg, 17
dram _dr, 21
electronmagneticmoment , 15
electronmass , 15
electronvolt _eV, 19
elementarycharge , 15
eulersnumber , 15
fahrenheit _degF, 21
farad _F, 16
fathom _ftm, 20
fermi _fermi, 18
finestructureconstant , 15
firkin _fir, 21
fluiddram _fl_dr, 20
fluidounce _fl_oz, 20
foot _ft, 20
fortnight _ftn, 21
furlong _fur, 20
gallon _gal, 20
gill _gi, 20
gradian _gon, 18
gram _g, 21
gramofTNT _g_TNT, 18
gravitationalconstant , 15
gray _Gy, 16
hand _hh, 20
hectare _hectare, 17
henry _H, 16
hertz _Hz, 16
horsepower _hp, 21
hour _h, 17
hundredweight _cwt, 21
inch _in, 20
intbritishthermalunit _BTU_it, 21
internationalcalorie _cal_IT, 18
joule _J, 16
katal _kat, 16
kilopond _kp, 18
knot _kn, 20
league _lea, 20
lightsecond _ls, 18
lightyear _ly, 18
liter _L, 17
longton _ton, 21
lumen _lm, 16
lux _lx, 16
metrichorsepower _PS, 19
metrictablespoon _Tbsp, 18
metricteaspoon _tsp, 18
mile _mi, 20
millimeterofmercury _mmHg, 18
<table>
<thead>
<tr>
<th>Currency</th>
<th>Symbol</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghan Afghani (AFN)</td>
<td>AFN</td>
<td>23</td>
</tr>
<tr>
<td>Afghan Pul (cAFN)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Albanian Lek (ALL)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Algerian Dinar (DZD)</td>
<td>DZD</td>
<td>24</td>
</tr>
<tr>
<td>Algerian Santeem (cDZD)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Angolan Centimo (cAOA)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Angolan Kwanza (cAOA)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Argentine Centavo (cARS)</td>
<td>ARS</td>
<td>23</td>
</tr>
<tr>
<td>Argentine Peso (ARS)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Armenian Dram (cAMD)</td>
<td>AMD</td>
<td>23</td>
</tr>
<tr>
<td>Armenian Luma (cAMD)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Aruban Cent (cAWG)</td>
<td>AWG</td>
<td>31</td>
</tr>
<tr>
<td>Aruban Florin (AWG)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Australian Cent (cAUD)</td>
<td>AUD</td>
<td>23</td>
</tr>
<tr>
<td>Australian Dollar (AUD)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Azerbaijani Manat (cAZN)</td>
<td>AZN</td>
<td>23</td>
</tr>
<tr>
<td>Azerbaijani Qepik (cAZN)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Bahamian Cent (cBSD)</td>
<td>BSD</td>
<td>31</td>
</tr>
<tr>
<td>Bahamian Dollar (BSD)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bahraini Dinar (BHD)</td>
<td>BHD</td>
<td>31</td>
</tr>
<tr>
<td>Bahraini Fils (cBHD)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bangladeshi Poisha (cBDT)</td>
<td>BDT</td>
<td>23</td>
</tr>
<tr>
<td>Bangladeshi Taka (BDT)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Barbadian Cent (cBBD)</td>
<td>BBD</td>
<td>31</td>
</tr>
<tr>
<td>Barbadian Dollar (BBD)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Belarusian Kupiejka (cBYN)</td>
<td>BYN</td>
<td>23</td>
</tr>
<tr>
<td>Belarusian Ruble (cBYN)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Belize Cent (cBZD)</td>
<td>BZD</td>
<td>31</td>
</tr>
<tr>
<td>Belize Dollar (BZD)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bermudian Cent (cBMD)</td>
<td>BMD</td>
<td>31</td>
</tr>
<tr>
<td>Bermudian Dollar (BMD)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bhutanese Chhertum (cBTN)</td>
<td>BTN</td>
<td>31</td>
</tr>
<tr>
<td>Bhutanese Ngultrum (BTN)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bolivian Boliviano (BOB)</td>
<td>BOB</td>
<td>23</td>
</tr>
<tr>
<td>Bolivian Centavo (cBOB)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Bosnian Fenings (cBAM)</td>
<td>BAM</td>
<td>23</td>
</tr>
<tr>
<td>Bosnian Mark (cBAM)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Botswana Pula (cBWP)</td>
<td>BWP</td>
<td>23</td>
</tr>
<tr>
<td>Botswana Thebe (cBWP)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Brazilian Centavo (cBRL)</td>
<td>BRL</td>
<td>23</td>
</tr>
<tr>
<td>Brazilian Real (BRL)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Brunei Dollar (BND)</td>
<td>BND</td>
<td>31</td>
</tr>
<tr>
<td>Brunei Sen (cBND)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Bulgarian Lev (BGN)</td>
<td>BGN</td>
<td>31</td>
</tr>
<tr>
<td>Bulgarian Stotinka (cBGN)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Burmese Kyat (MKK)</td>
<td>MKK</td>
<td>27</td>
</tr>
<tr>
<td>Burmese Pya (cMKK)</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>Burundian Centime (cBIF)</td>
<td>BIF</td>
<td>23</td>
</tr>
<tr>
<td>Burundian Franc (BIF)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Cambodian Riel (KHR)</td>
<td>KHR</td>
<td>26</td>
</tr>
<tr>
<td>Canadian Cent (cCAD)</td>
<td>CAD</td>
<td>24</td>
</tr>
<tr>
<td>Canadian Dollar (CAD)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Cape Verdean Centavo (cCVE)</td>
<td>CVE</td>
<td>32</td>
</tr>
<tr>
<td>Cape Verdean Escudo (cCVE)</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Cayman Islands Cent (cKYD)</td>
<td>KYD</td>
<td>32</td>
</tr>
<tr>
<td>Cayman Islands Dollar (KYD)</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Central African CFA Centime (cXAF)</td>
<td>XAF</td>
<td>33</td>
</tr>
<tr>
<td>Central African CFA Franc (cXAF)</td>
<td>XAF</td>
<td>33</td>
</tr>
<tr>
<td>CFP Centime (cXPF)</td>
<td>XPF</td>
<td>34</td>
</tr>
<tr>
<td>CFP Franc (cXPF)</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Chilean Centavo (cCLP)</td>
<td>CLP</td>
<td>24</td>
</tr>
<tr>
<td>Chilean Peso (CLP)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Chinese Renminbi Fen (cCNY)</td>
<td>CNY</td>
<td>24</td>
</tr>
<tr>
<td>Chinese Renminbi Yuan (cCNY)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Colombian Centavo (cCOP)</td>
<td>COP</td>
<td>24</td>
</tr>
<tr>
<td>Colombian Peso (COP)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Comorian Centime (cKMF)</td>
<td>KMF</td>
<td>32</td>
</tr>
<tr>
<td>Comorian franc (cKMF)</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Congolese Centime (cCDF)</td>
<td>CDF</td>
<td>24</td>
</tr>
<tr>
<td>Congolese Franc (CDF)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Costa Rican Centimos (cCRC)</td>
<td>CRC</td>
<td>24</td>
</tr>
<tr>
<td>Costa Rican Colon (CRC)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Croatian Kuna (HRK)</td>
<td>HRK</td>
<td>25</td>
</tr>
<tr>
<td>Croatian Lira (CHRK)</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Cuban Centavo (cCUP)</td>
<td>CUP</td>
<td>31</td>
</tr>
<tr>
<td>Cubano Convertible Centavo (cCUC)</td>
<td>CUC</td>
<td>31</td>
</tr>
<tr>
<td>Cubano Convertible Peso (cCUC)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Cuban Peso (CUP)</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Czech Haler (cCZK)</td>
<td>CZK</td>
<td>24</td>
</tr>
<tr>
<td>Czech Koruna (cCZK)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Currency Name</td>
<td>Code</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Danish Krone (DKK)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Danish Ore (cDKK)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Djiboutian Centime (cDJF)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Djiboutian Franc (DJF)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Dominican Centavo (cDOP)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Dominican Peso (cDOP)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Eastern Caribbean Cent (cXCD)</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Eastern Caribbean Dollar (XCD)</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Egyptian Piastre (cEGP)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Egyptian Pound (EGP)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Eritrean Cent (cERN)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Eritrean Nakfa (ERN)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Ethiopian Birr (ETB)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Ethiopian Santim (cETB)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Falkland Islands Penny (cFKP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Falkland Islands Pound (FKP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Fijian Cent (cFJD)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Fijian Dollar (FJD)</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Gambian Butut (cGMD)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Gambian Dalasi (GMD)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Georgian Lari (GEL)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Georgian Tetri (cGEL)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Ghanaian Cedi (GHS)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Ghanaian Pesewa (cGHS)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Gibraltar Penny (cGIP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Gibraltar Pound (GIP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Guatemalan Centavo (cGTQ)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Guatemalan Quetzal (GTQ)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Guernsey Penny (cGGP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Guernsey Pound (GGP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Guinean Centime (cGNF)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Guinean Franc (GNF)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Guyanese Cent (cGYY)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Guyanese Dollar (GYY)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Haitian Centime (cHTG)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Haitian Gourde (HTG)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Honduran Centavo (cHN)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Honduran Lempira (HNL)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Hong Kong Cent (cHKD)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Hong Kong Dollar (HKD)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Hungarian Filler (cHUF)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Hungarian Forint (HUF)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Icelandic Krone (ISK)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Indian Paisa (cINR)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Indian Rupee (INR)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Indonesian Rupiah (cIDR)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Indonesian Sen (cIDR)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Iranian Rial (IRR)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Iranian Toman (cIRR)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Iraqi Dinar (cIQD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Iraqi Fils (cIQD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Israeli New Agora (cILS)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Israeli New Shekel (ILS)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Jamaican Cent (cJMD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Jamaican Dollar (JMD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Japanese Yen (JPY)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Jersey Penny (cJEP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Jersey Pound (JEP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Jordanian Dinar (cJOD)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Jordanian Fils (cJOD)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Kazakhstani Tenge (KZT)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kazakhstani Tiyin (cKZT)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kenyan Cent (cKES)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kenyan Shilling (KES)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kiribati Cent (cKID)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Kiribati Dollar (KID)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Kuwaiti Dinar (KWD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kuwaiti Fils (cKWD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kyrgyzstani Som (KGS)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Kyrgyzstani Tiyin (cKGS)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Lao Att (cLAK)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Lao Kip (LAK)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Lebanese Pound (cLBP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Lebanese Qeresh (cLBP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Liberian Cent (cLRD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Liberian Dollar (LRD)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Libyan Dinar (cLYD)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Libyan Dirham (cLDY)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Macanese Avo (cMOP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Macanese Pataca (MOP)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Macedonian Denar (cMKD)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Macedonian Deni (cMKD)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Malagasy Ariary (cMGA)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Malagasy Iraimbilanja (cMGA)</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Currency Name</td>
<td>Code</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Somaliland Cent</td>
<td>cSOS</td>
<td>29</td>
</tr>
<tr>
<td>Somaliland Shilling</td>
<td>_cSQS</td>
<td>29</td>
</tr>
<tr>
<td>Somali Senti</td>
<td>_cSOS</td>
<td>29</td>
</tr>
<tr>
<td>Somali Shilling</td>
<td>_SOS</td>
<td>29</td>
</tr>
<tr>
<td>South African Cent</td>
<td>cZAR</td>
<td>30</td>
</tr>
<tr>
<td>South African Rand</td>
<td>_ZAR</td>
<td>30</td>
</tr>
<tr>
<td>South Korean Jeon</td>
<td>_cKRW</td>
<td>26</td>
</tr>
<tr>
<td>South Korean Won</td>
<td>_KRW</td>
<td>26</td>
</tr>
<tr>
<td>South Sudanese Piaster</td>
<td>_cSSP</td>
<td>33</td>
</tr>
<tr>
<td>South Sudanese Pound</td>
<td>_SSP</td>
<td>33</td>
</tr>
<tr>
<td>Sri Lankan Cent</td>
<td>_cLKR</td>
<td>26</td>
</tr>
<tr>
<td>Sri Lankan Rupee</td>
<td>_LKR</td>
<td>26</td>
</tr>
<tr>
<td>Sudanese Pound</td>
<td>_SDG</td>
<td>29</td>
</tr>
<tr>
<td>Sudanese Qirsh</td>
<td>_cSDG</td>
<td>29</td>
</tr>
<tr>
<td>Suriname Cent</td>
<td>_cSRD</td>
<td>29</td>
</tr>
<tr>
<td>Suriname Dollar</td>
<td>_SRD</td>
<td>29</td>
</tr>
<tr>
<td>Swazi Cent</td>
<td>_cSZL</td>
<td>33</td>
</tr>
<tr>
<td>Swazi Lilangeni</td>
<td>_SZL</td>
<td>33</td>
</tr>
<tr>
<td>Swedish Krona</td>
<td>_SEK</td>
<td>29</td>
</tr>
<tr>
<td>Swedish Ore</td>
<td>_cSEK</td>
<td>29</td>
</tr>
<tr>
<td>Swiss Franc</td>
<td>_cCHF</td>
<td>24</td>
</tr>
<tr>
<td>Swiss Rappen</td>
<td>_cCHF</td>
<td>24</td>
</tr>
<tr>
<td>Syrian Piaster</td>
<td>_cSYP</td>
<td>29</td>
</tr>
<tr>
<td>Syrian Pound</td>
<td>_cSYR</td>
<td>29</td>
</tr>
<tr>
<td>Tajikistani Diram</td>
<td>_cTJS</td>
<td>29</td>
</tr>
<tr>
<td>Tajikistani Somani</td>
<td>_TJS</td>
<td>29</td>
</tr>
<tr>
<td>Tanzanian Senti</td>
<td>_cTZS</td>
<td>30</td>
</tr>
<tr>
<td>Tanzanian Shilling</td>
<td>_TZS</td>
<td>30</td>
</tr>
<tr>
<td>Thai Baht</td>
<td>_THB</td>
<td>29</td>
</tr>
<tr>
<td>Thai Satang</td>
<td>_cTHB</td>
<td>29</td>
</tr>
<tr>
<td>Tongan Paanga</td>
<td>_TOP</td>
<td>29</td>
</tr>
<tr>
<td>Tongan Senti</td>
<td>_cTOP</td>
<td>29</td>
</tr>
<tr>
<td>Transnistrian Kopeck</td>
<td>_cPRB</td>
<td>33</td>
</tr>
<tr>
<td>Transnistrian Ruble</td>
<td>_cPRB</td>
<td>33</td>
</tr>
<tr>
<td>Trinidad And Tobago Cent</td>
<td>_cTTD</td>
<td>30</td>
</tr>
<tr>
<td>Trinidad And Tobago Dollar</td>
<td>_TTD</td>
<td>30</td>
</tr>
<tr>
<td>Turkish Kurus</td>
<td>_cTRY</td>
<td>30</td>
</tr>
<tr>
<td>Turkish Lira</td>
<td>_TRY</td>
<td>30</td>
</tr>
<tr>
<td>Turkmenistan Manat</td>
<td>_cTMT</td>
<td>33</td>
</tr>
<tr>
<td>Turkmenistan Tenge</td>
<td>_cTMT</td>
<td>33</td>
</tr>
<tr>
<td>Tuvaluan Cent</td>
<td>_cTVD</td>
<td>33</td>
</tr>
<tr>
<td>Tuvaluan Dollar</td>
<td>_TVD</td>
<td>33</td>
</tr>
<tr>
<td>Ugandan Cent</td>
<td>_cUGX</td>
<td>30</td>
</tr>
<tr>
<td>Ugandan Shilling</td>
<td>_UGX</td>
<td>30</td>
</tr>
<tr>
<td>Ukrainian Hryvnia</td>
<td>_UAH</td>
<td>30</td>
</tr>
<tr>
<td>Ukrainian Kopiyka</td>
<td>_cUAH</td>
<td>30</td>
</tr>
<tr>
<td>United Arab Emirates Dirham</td>
<td>_AED</td>
<td>31</td>
</tr>
<tr>
<td>United Arab Emirates Fils</td>
<td>_cAED</td>
<td>31</td>
</tr>
<tr>
<td>Uruguayan Centesimo</td>
<td>_cUYU</td>
<td>30</td>
</tr>
<tr>
<td>Uruguayan Peso</td>
<td>_UYU</td>
<td>30</td>
</tr>
<tr>
<td>US Cent</td>
<td>_cUSD</td>
<td>30</td>
</tr>
<tr>
<td>US Dollar</td>
<td>_USD</td>
<td>30</td>
</tr>
<tr>
<td>Uzbekistani Som</td>
<td>_cUZS</td>
<td>30</td>
</tr>
<tr>
<td>Uzbekistani Tyin</td>
<td>_cUZS</td>
<td>30</td>
</tr>
<tr>
<td>Venezuelan Bolivar Soberano</td>
<td>_cVES</td>
<td>30</td>
</tr>
<tr>
<td>Venezuelan Centimo Soberano</td>
<td>_cVES</td>
<td>30</td>
</tr>
<tr>
<td>Vietnamese Dong</td>
<td>_cVND</td>
<td>30</td>
</tr>
<tr>
<td>Vietnamese Xu</td>
<td>_cVND</td>
<td>30</td>
</tr>
<tr>
<td>West African CFA Centime</td>
<td>_cXOF</td>
<td>34</td>
</tr>
<tr>
<td>West African CFA Franc</td>
<td>_cXOF</td>
<td>34</td>
</tr>
<tr>
<td>Yemeni Dinar</td>
<td>_cYER</td>
<td>30</td>
</tr>
<tr>
<td>Yemeni Rial</td>
<td>_cYER</td>
<td>30</td>
</tr>
<tr>
<td>Zambian Kwacha</td>
<td>_cZMW</td>
<td>30</td>
</tr>
<tr>
<td>Zambian Ngwee</td>
<td>_cZMW</td>
<td>30</td>
</tr>
<tr>
<td>Zimbabwean Bonds</td>
<td>_cZWL</td>
<td>34</td>
</tr>
<tr>
<td>Zimbabwean Cent</td>
<td>_cZWL</td>
<td>34</td>
</tr>
</tbody>
</table>
Index of Lua Classes and Methods

D.defineBase(symbol, name), 46
D.new(d), 46
N.abs(n), 49
N.acos(n), 52
N.asin(n), 52
N.atan(n), 53
N.cos(n), 51
N.cosh(q), 53
N.exp(n), 51
N.log(n, base=nil), 50
N.mean(n), 49
N.new(n=nil), 48
N.new(str), 48
N.new(x, dx=nil), 48
N.sin(n), 51
N.sinh(q), 53
N.sqrt(n), 50
N.tan(n), 52
N.tanh(q), 54
N.uncertainty(n), 49
Q.abs(q), 39
Q.acos(q), 42
Q.acosh(q), 44, 55
Q.addPrefix(prefixes, units), 37
Q.asin(q), 42
Q.asinh(q), 44, 54
Q.atan(q), 42
Q.atanh(q), 44, 55
Q.cos(q), 41
Q.cosh(q), 43
Q.log(q, base=nil), 40
Q.max(q1, q2, ...), 39
Q.min(q1, q2, ...), 39
Q.new(q=nil), 35
Q.exp(q), 40
Q.isclose(self, q, r), 37
Q.log(q, base=nil), 40
Q.atan(q), 42
Q.acosh(q), 44, 55
Q.addPrefix(symbol, name, q), 36
Q.defineBase(symbol, name, dimension), 35
Q.definePrefix(symbol, name, factor), 37
Q.sin(q), 41
Q.sinh(q), 43
Q.sqrt(q), 40
Q.tan(q), 41
Q.tanh(q), 43
Q.to(self, q=nil), 38
Q.tosiunitx(self, param, mode=Q.siunitx_SI), 38
Unit.new(symbol, name, prefixsymbol=nil, prefixname=nil), 47
Unit.new(u=nil), 47
Unit.tosiunitx(self), 47