PRPN Quick Guide

how to enter numbers:
A number is erverytime beginning with a digit. A negative number can be entered with a leading understroke, as the minus sign is an operator.
Examples:

type in:	for:
`1`	1
`_1`	-1
`1.5`	1.5
`_0.3`	-0.3

After each entry press the `ENTER` key. There are only a very few commands where it is not necessary to press the `ENTER` key before:

`+ - / *` and `^`

You might use powers of ten by adding an `e` or `E`.
Examples:

type in:	for:
`1.5e-6`	$1.5 \cdot 10^{-6}$
`1.5E-6`	$1.5 \cdot 10^{-6}$

Even a number with error can be typed in by using the hash (`#`) as the $\pm$. Please note: both numbers, before and behind the hash, can have an power of ten.
The error is calculated with the Gaussian error propagation, which is included for nearly all operations now. But please be carefull with this feature, as it is not proved well now.
Examples:

type in:	for:
`100#0.2`	100.0 ± 0.2
`1e2#2e-1`	100.0 ± 0.2

Entering units is done by using a understroke between number and unit:
first enter the number followed by an understroke and than followed by the unit. Please refer to the attached document, how to spell the specific units (many imperial or US units are followed by a dot, as it is a short form, though some of them might be written without a dot).
Examples:

type in:	for:
`1_kg`	$1\,kg$
`1_m`	$1\,m$
`1_s`	$1\,s$
`1_V`	$1\,V$
`1_in`	$1\,in$
`1_in.`	$1\,in$
`1_gal.`	$1\,gal_{imp}$
`1_us.gal.`	$1\,gal_{US}$

For SI units the letters for the prefactors might be used in front of the units. All prefactors from yotta (10^{24}) to yocto (10^{-24}) can be used. The micro can be written as an `u` or as an μ. This is available for SI conform units only!
Examples:

type in:	for:
`1_um`	$1\,\mu m$
`1_µm`	$1\,\mu m$
`1_MV`	$1\,MV$
`1_nF`	$1\,nF$

Powers of units can be entered by using the `^` sign. The exponent might be either positive or negative, even floating point numbers are allowed.
Examples:

type in:	for:
`1_m^2`	$1\,m^2$
`1_s^-1`	$1\,\frac{1}{s}$
`1_Hz^0.5`	$1\,\sqrt{Hz}$

Units can be multiplied by using the asterisk sign $*$.
Examples:

type in:	for:
`1_kg*m`	$1\,kg \cdot m$
`1_in*ft^2`	$1\,in \cdot ft^2$

Units can be divided by using the slash sign `/`.
Note: any unit after the 1st slash stands in the nominator, independend if the is combined via a slash or an asterisk!
Also note: do not brackets at all while entering units!
Examples:

type in:	for:
`1_m/s`	$1\,\frac{m}{s}$
`1_kg*m/s^2`	$1\,\frac{kg \cdot m}{s^2}$
`1_kg*m^1/s^2*A^2`	$1\,\frac{kg \cdot m^2}{s^2 \cdot A^2}$
`1_kg*m^1/s^2/A^2`	$1\,\frac{kg \cdot m^2}{s^2 \cdot A^2}$

If you want to enter a number with error, place the unit at the end of the entry:
Examples:

type in:	for:
`1e2#2e-1_W/K*m`	$100.0\,\frac{W}{K \cdot m} \ \pm \ 0.2\,\frac{W}{K \cdot m}$

how to convert units:
If the shown units is not comfortable, it can be converted into each unit of the same dimension by using `>_` in front of the units. If one unit has an error (like electron volts or the astronomical unit), this is automaticaly displayed.

given is:	type in:	to get:
$1\,\mu V/e^-$	`>_1/F`	$6.24150948E+12\,\frac{1}{F} \ \pm \ 5.5E+05\,\frac{1}{F}$

With `>_si` a units can converted to the SI base every time.

given is:	type in:	to get:
$1\,\mu V/e^-$	`>_si`	$6.24150948E+12\,\frac{kg \cdot m^2}{s^4 \cdot A^2} \ \pm \ 5.5E+05\,\frac{kg \cdot m^2}{s^4 \cdot A^2}$

With `>_+` it is possible to scroll through all known units of the same dimension step by step.
With `>_?` it is possible to scroll through all known units of the same dimension using the up and down arrow keys. The conversion of the unit is automatically shown in line two. The left and right arrow keys change the direction of conversion. Leave the function by pressing `<ENTER>`.
This function is not well tested yet!

The database includes all SI bases and some additional pseudo bases:

SI bases:

for:	type in:	to get:
mass	*kg*	kilo grams
length	*m*	meter
time	*s*	seconds
electrical current	*A*	Ampere
temperatur	*K*	Kelvin
lightforce	*cd*	candela
amount of substance, quanity	*mol*	Mol

SI certified pseudo bases:

for:	type in:	to get:
flat angle	*rad*	radian
solid angle	*sr*	steradiant

additional non SI certified pseudo bases:

for:	type in:	to get:
storage (computer)	*bit*	bit
currency	*EUR*	Euro
ratio	: 1	to one
logarythmic ratio, 10*lg	*dBm*	deci Bel milli watt
logarythmic ratio, 20*lg	*dBu*	deci Bel micro volt
pixel number	*pix*	pixel

special unit which gets replaced by the unit of the other operand:
(usefull for add and sub operations)

for:	type in:	to get:
auto replacement	*any*	the unit of the other operand

The following bases can be converted into each other:

1	↔	: 1	1	↔	*pix*						
: 1	↔	*rad*	: 1	↔	*sr*	*rad*	↔	*sr*			
: 1	↔	*dBm*	: 1	↔	*dBW*	*W*	↔	*dBm*	*W*	↔	*dBW*
: 1	↔	*dBu*	: 1	↔	*dBuV*	*V*	↔	*dBu*	*V*	↔	*dBuV*
: 1	↔	*dBV*	: 1	↔	*dBA*	*V*	↔	*dBV*	*Pa*	↔	*dBA*
dBm	↔	*dBW*	*dBm*	↔	*dBu*	*dBm*	↔	*dBuV*	*dBm*	↔	*dBV*
dBW	↔	*dBu*	*dBW*	↔	*dBuV*	*dBW*	↔	*dBV*			
dBu	↔	*dBuV*	*dBu*	↔	*dBV*	*dBuV*	↔	*dBV*			

Some units get modified by operations in the following way:

general rules:

°C	+	K	→	°C	°C	-	K	→	°C	°C	-	°C	→	K
°F	+	°Ra	→	°F	°F	-	°Ra	→	°F	°F	-	°F	→	°Ra
?*	+	*any*	→	?*	?*	-	*any*	→	?*	**(note: ?* = any unit)**				
?*	*	*any*	→	?*	?*	/	*any*	→	?*	**(note: ?* = any unit)**				

if `AUTO` is set:

?*	*	: 1	→	?*	?*	/	?*	→	: 1	**(note: ?* = any unit)**
pix	*	*pix*	→	*pix*	*pix*	/	*pix*	→	*pix*	

if `AUTORAD` is set:

rad	*	*m*	→	*m*	*sr*	*	m^2	→	m^2					
m	/	*m*	→	*rad*	m^2	/	m^2	→	*sr*					
W	/	*W*	→	*dBW*	*J*	/	*J*	→	*dBW*	*Pa*	/	*Pa*	→	*dBA*
V	/	*V*	→	*dBV*	*C*	/	*C*	→	*dBV*	*A*	/	*A*	→	*dBV*

nature constants can be used by a leading backslash:

for:	type in:	to get:
speed of light	\c	299792458_m/s
elementary charge	\e	1.60217653e-19#0.00000014e-19_C
Planck constant	\h	6.6260693e-34#0.0000011e-34_J*s
Planck constant by 2 PI	\hbar	1.054571682E-34#1.160028851E-74_J*s
Boltzmann constant	\k	13.80650388#2.5E-05_μJ/K
Avogadro constant	\Na	6.0221415e23#0.0000010e23_1/mol
molar gas constant	\R	8.314472#0.000015_J/mol*K
molar volume of ideal gas	\V_m	22.413996E-3#0.000039E-3_m^3/mol
magnetic constant	\mu0	1.256637061_μH/m
electric constant	\e0	8.854187818_pF/m
impedance of vacuum	\Z0	376.7303135_Ohm
constant of gravitation	\G	6.6742e-11#0.0010e-11_m^3/kg*s^2
standard acceleration of gravity	\g	9.80665_m/s^2
standard atmosphere	\atm	101325_Pa
electron mass	\m_e	9.1093826e-31#0.0000016e-31_kg
electron magn. moment	\mu_e	-928.476412e-26#0.000080e-26_J/T
proton mass	\m_p	1.67262171e-27#0.00000029e-27_kg
proton magn. moment	\mu_p	1.41060671e-26#0.00000012e-26_J/T
proton rms charge radius	\R_p	0.8750e-15#0.0068e-15_m

For a complete list take a look into the file `/usr/local/lib/prpn-0.2/constants/const.txt`**.**

This is an incomplete list of commands which are known by prpn:

- **standard fuctions:**

`+`	**add**	$x_1 = x_2 + x_1$
`-`	**substrate**	$x_1 = x_2 - x_1$
`*`	**multiply**	$x_1 = x_2 * x_1$
`/`	**devide**	$x_1 = x_2/x_1$
`^`	**power**	$x_1 = x_2^{x_1}$

- **powers and logarithm:**

`pow <ENTER>`	**power**	$x_1 = x_2^{x_1}$
`sq <ENTER>`	**sqare**	$x_1 = x_1^2$
`root <ENTER>`	**root**	$x_1 = \sqrt[x_1]{x_2}$
`sqrt <ENTER>`	**square root**	$x_1 = \sqrt{x_1}$
`log <ENTER>`	**logarithm**	$x_1 = log_{x_2}(x_1)$
`ln <ENTER>`	**logarithm to base e**	$x_1 = log_e(x_1)$
`lg <ENTER>`	**logarithm to base 10**	$x_1 = log_{10}(x_1)$

- **trigonometric functions:**

`sin <ENTER>`	**sine**	$x_1 = sin(x_1)$
`cos <ENTER>`	**cosine**	$x_1 = cos(x_1)$
`sec <ENTER>`	**secans**	$x_1 = sec(x_1) = \frac{1}{cosx_1}$
`tan <ENTER>`	**tangens**	$x_1 = tan(x_1)$
`cot <ENTER>`	**cotangens**	$x_1 = cot(x_1)$
`asin <ENTER>`	**arcus sine**	$x_1 = asin(x_1)$
`acos <ENTER>`	**arcus cosine**	$x_1 = acos(x_1)$
`asec <ENTER>`	**arcus secans**	$x_1 = asec(x_1)$
`atan <ENTER>`	**arcus tangens**	$x_1 = atan(x_1)$
`acot <ENTER>`	**arcus cotangens**	$x_1 = acot(x_1)$

- **hyperbolic functions:**

`sinh <ENTER>`	**hyperbolic sine**	$x_1 = sinh(x_1)$
`cosh <ENTER>`	**hyperbolic cosine**	$x_1 = cosh(x_1)$
`tanh <ENTER>`	**hyperbolic tangens**	$x_1 = tanh(x_1)$
`asinh <ENTER>`	**arcus hyperbolic sine**	$x_1 = asinh(x_1)$
`acosh <ENTER>`	**arcus hyperbolic cosine**	$x_1 = acosh(x_1)$
`atanh <ENTER>`	**arcus hyperbolic tangens**	$x_1 = atanh(x_1)$

- **display modes:**

`std <ENTER>`	**selects standard floating point display mode**
`n <ENTER> fix <ENTER>`	**n = 0 to 15, selects the fixed decimal place display mode**
`n <ENTER> sci <ENTER>`	**n = 0 to 15, selects the scientific fixed decimal place display mode which uses exponents all the time**
`n <ENTER> eng <ENTER>`	**n = 0 to 15, selects the engineering fixed decimal place display mode which uses intervals of** 10^3
`n <ENTER> tec <ENTER>`	**n = 0 and 2 to 15, selects the technican (fixed) decimal place display mode, uses typical non SI units, like litre, hour, etc. A parameter of 0 sets the floating point display mode.**
`deg <ENTER>`	**displays angles in floting point degrees**
`dms <ENTER>`	**displays angles in degrees minutes and seconds**
`rad <ENTER>`	**displays angles in radian**
`gon <ENTER>`	**displays angles in new degree**

- **others:**

up and down arrow keys	**select the line to be duplicated or droped**
`dup <ENTER>`	**duplicates the selected entry, a single** `<ENTER>` **at empty input line will do the same**
`drop <ENTER>`	**drops the selected entry, a single** `<BACKSPACE>` **at empty input line will do the same**
`clr <ENTER>`	**clears the complete stack**
`rpn <ENTER>`	**selects the RPN mode**
`alg <ENTER>`	**selects the algebraic mode (for those who need this)**