Using Units and Physical Constants in Mathematica

Mathematica has the ability to handle many calculations with units, and to perform conversions among different units. It also has many built-in constants and other data that are useful in doing calculations.

```
In[54]:= Clear["Global`*"]
```

Basic Units

Imagine finding the volume of a box (2m) x (3m) x (5 cm). (Note the mixed units.) Enter the quantities in Mathematica using the 'Quantity' function. Mathematica will automatically suggest some units once you start typing. Note that units in Mathematica are always plural.

```
In[55]:= l = Quantity[2., "Meters"]
Out[55]=
 2. m
```

```
In[56]:= w = Quantity[3., "Meters"]
Out[56]=
 3. m
```

```
In[57]:= h = Quantity[5., "Centimeters"]
Out[57]=
 5. cm
```

```
In[58]:= volume = l * w * h
Out[58]=
```

$0.3\ m^3$

You can convert to other volume units with UnitConvert.

```
In[59]:= UnitConvert[volume, "Liters"]
Out[59]=
 300. L
```

```
In[60]:= UnitConvert[volume, "Centimeters"^3]
Out[60]=
```

$300\,000.\ cm^3$

```
In[61]:= UnitConvert[volume, "Gallons"]
Out[61]=
 79.2516 gal
```

If you just want to convert to SI units, you can simply use "SI":

In[62]:=
```
UnitConvert[volume, "SI"]
```
Out[62]=

$0.3\ m^3$

Physical Constants

Mathematica has many constants built in, though sometimes you have to guess exactly how it will be expressed. For example, Boltzmann's constant k is:

In[63]:=
```
k = Quantity[1, "BoltzmannConstant"]
```
Out[63]=

k

Mathematica simply writes the standard letter *k* for this constant. To see what it is numerically, use UnitConvert[ ]:

In[64]:=
```
UnitConvert[k, "SI"]
```
Out[64]=

$$\frac{1\,380\,649}{16\,021\,766\,340}\ \mathrm{eV/K}$$

There are two quirks with this answer. First, the set of units displayed is not always the most useful. If you know the units you want, you can request them directly. If you simply want the traditional SI base units (kg, m, s, A, K), use "SIBase".

In[65]:=
```
UnitConvert[k, "Joules" / "Kelvins"]
UnitConvert[k, "SIBase"]
```
Out[65]=

$$\frac{1\,380\,649}{100\,000\,000\,000\,000\,000\,000\,000\,000\,000}\ \mathrm{J/K}$$

Out[66]=

$$\frac{1\,380\,649}{100\,000\,000\,000\,000\,000\,000\,000\,000\,000}\ \mathrm{kg\,m^2/(s^2 K)}$$

The second quirk is that for constants defined exactly in SI (such as the speed of light or Boltzmann's constant) Mathematica returns an exact integer result. You usually want to convert that to a numerical approximation with the N[ ] function.

In[67]:=
```
UnitConvert[N[k], "Joules" / "Kelvins"]
```
Out[67]=

$1.38065 \times 10^{-23}\ \mathrm{J/K}$

Physical and Chemical Data

Mathematica has access to a wealth of data. Some is built in, but most requires an on-line connection to fetch the data from Wolfram's servers.

For example, data about the elements is available through the ElementData function. Here is how to get the mass of a helium atom. The result is returned in terms of the atomic mass unit, but you can convert it to other forms:

In[68]:=
```
m = ElementData["Helium", "AtomicMass"]
```
Out[68]=
```
4.002602 u
```

In[69]:=
```
UnitConvert[m, "Kilograms"]
```
Out[69]=

6.646477×10^{-27} kg

In[70]:=
```
UnitConvert[m, "Megaelectronvolts" / "SpeedOfLightSquared"]
```
Out[70]=

3728.400 MeV/c^2

You can get a list of all available properties by using the “Properties” key:

In[71]:=
```
ElementData["Helium", "Properties"]
```

Out[71]=

{adiabatic index, allotropes, allotropic multiplicities, alternate names, atomic mass, atomic number, atomic radius, atomic symbol, block, boiling point T_b, Brinell hardness, bulk modulus, CAS number, color, common compound names, covalent radius, critical pressure p_c, critical temperature T_c, crust abundance, crust molar abundance, Bravais lattice, lattice schematic, Curie point, Debye characteristic temperature, decay mode, discoverers, discovery country, discovery date, electrical conductivity, electrical type, electron affinity, electron count, electronegativity, electronic configuration, electron shell configuration, entity classes, entity type list, molar heat of fusion $\Delta_{fus}H$, gas atomic multiplicities, group, half-life, human abundance, human molar abundance, icon color, image, ionization energies, isotope abundances, isotope half-lives, isotope lifetimes, known isotopes, known oxidation states, lattice angles, lattice constants, Lewis structure, liquid density, magnetic type, mass density, mass magnetic susceptibility, melting point T_m, memberships, meteorite abundance, meteorite molar abundance, Mohs hardness, molar heat capacity, molar magnetic susceptibility, molar mass, molar radioactivity, molar volume, most common oxidation states, name, Néel point, number of neutrons per isotope, neutron cross-section, neutron mass absorption, nuclear diameter, nuclear radius, ocean abundance, ocean molar abundance, orbital occupation list, origins, period, phase at STP, Poisson ratio, price, proton count, refractive index, resistivity, series, shear modulus, short electronic configuration, solar abundance, solar molar abundance, molar heat of solidification, sound speed, space group, IUCr space group number, specific heat of fusion $\Delta_{fus}h$, specific heat capacity c_p, specific ionization energies, specific radioactivity, specific heat of solidification, stable isotopes, superconducting point, tensile yield strength, term symbol, thermal conductivity, thermal expansion, threshold frequency, ultimate tensile strength, universe abundance, universe molar abundance, unstable isotopes, valence, number of valence electrons, van der Waals radius, molar heat of vaporization $\Delta_{vap}H$, Vickers hardness, volume magnetic susceptibility, work function, Young modulus}

Data about chemical compounds is available through the ChemicalData[ ] function. For example, the mass of the water molecule is (where ‘u’ is the Atomic Mass Constant)

In[72]:=
```
mH2O = ChemicalData["Water", "MolecularMass"]
```

Out[72]=

18.015 u

In[73]:=
```
Quantity[1, "AtomicMassConstant"]
```
Out[73]=

m_u

In[74]:=
```
UnitConvert[%, "Kilograms"]
```
Out[74]=

$1.66053907 \times 10^{-27}$ kg

Calculations

Mathematica can do calculations with units and will perform appropriate operations on units. It will refuse to do calculations where the units do not agree.

Room Temperature

In[75]:=
```
k = Quantity[1, "BoltzmannConstant"]
```
Out[75]=

k

In[76]:=
```
T = Quantity[300., "Kelvins"]
```
Out[76]=

300. K

In[77]:=
```
UnitConvert[k T, "Joules"]
```
Out[77]=

4.14195×10^{-21} J

In[78]:=
```
UnitConvert[k T, "Electronvolts"]
```
Out[78]=

0.025852 eV

In[79]:=
```
k + T
```
··· Quantity : Kelvins and BoltzmannConstant are incompatible units

··· Quantity : Kelvins and BoltzmannConstant are incompatible units

Out[79]=

k + 300. K

For a longer example, consider problem 2.35. The final step is calculating a temperature from a complex combination of constants. In this problem, we are considering Helium initially at “room temperature” and “atmospheric pressure”.

In[85]:=
```
Clear[Tz, h, k, m, Ti, pi]
```

In[86]:=
```
Tz = h^2 / (2 ⅇ^(5/3) k m π (k Ti / pi)^(2/3))
```

Out[86]=

$$\frac{h^2}{2\, e^{5/3}\, k\, m\, \pi \left(\frac{k\, Ti}{pi}\right)^{2/3}}$$

In[87]:=
```
Ti = Quantity[300, "Kelvins"]
```

Out[87]=

300 K

In[88]:=
```
pi = Quantity[1, "Atmospheres"]
```

Out[88]=

1 atm

In[89]:=
```
k = Quantity[1, "BoltzmannConstant"]
```

Out[89]=

k

In[90]:=
```
h = Quantity[1, "PlanckConstant"]
```

Out[90]=

h

In[91]:=
```
m = ElementData["Helium", "AtomicMass"]
```

Out[91]=

4.002602 u

In[92]:=
```
Tz
```

Out[92]=

$0.0001675866\ atm^{2/3} h^2 / (u\, K^{2/3} k^{5/3})$

In[93]:=
```
UnitConvert[Tz, "Kelvins"]
```

Out[93]=

0.01212015 K