```
Conversion factor from parsecs to meters:
>>> from astropy import units as u
>>> u.pc.to(u.m)
3.0856776e+16

Conversion factor from km/s to pc/yr
>>> (u.km / u.s).to(u.pc / u.Myr)
1.0227121650537077

Convert SI units for density to the c.g.s system:
>>> (u.kg / u.m**3).to_system(u.cgs)
[Unit("0.001 g / cm3")]

Decompose Newtons into base SI units:
>>> u.N.decompose()
Unit("kg m / s2")

Find higher-level units equivalent to kg.m/s^2:
>>> (u.kg * u.m / u.s ** 2).compose()
[Unit("N"), Unit("100000 dyn")]

Convert wavelength to frequency:
>>> u.nm.to(u.GHz, 1000., equivalencies=u.spectral())
299792.458

Define a quantity from scalars and units:
>>> from astropy import units as u
>>> 15.1 * u.m / u.s
<Quantity 15.1 m / s>

Convert a distance:
>>> (1.15e13 * u.km).to(u.pc)
<Quantity 0.372689618289 pc>

Make use of the unit equivalencies:
>>> e = 130. * u.eV
>>> e.to(u.Angstrom, equivalencies=u.spectral())
<Quantity 95.37245609234003 Angstrom>

Combine quantities:
>>> x = 1.4E11*u.km / (0.7*u.Myr) / (4.1E11*u.s)
<Quantity 0.487804878049 km / (Myr s)>

Convert to SI units:
>>> x.decompose()
<Quantity 1.54579339587e-11 m / (s2)>

Access physical constants:
>>> from astropy.constants import G, M_sun
>>> print G
  Name   = Gravitational constant
  Value  = 6.67384e-11
  Error  = 8e-15
  Units  = m3 / (kg s2)
  Reference = CODATA 2010

Combine quantities and constants:
>>> F = G * (3 * M_sun) * (2 * u.kg) / (1.5 * u.au) ** 2
>>> F.to(u.N)
<Quantity 0.01581795428812989 N>
```