Tricks

1. Use `==` to test for equality. In effect, `A==B` is equivalent to `simplify(A-B)==0`.

2. In a script, line breaking is allowed where the scanner needs something to complete an expression. For example, the scanner will automatically go to the next line after an operator.

3. Setting `trace=1` in a script causes each line to be printed just before it is evaluated. Useful for debugging.

4. The last result is stored in symbol `last`.

5. Use `contract(A)` to get the mathematical trace of matrix A.

6. Use `binding(s)` to get the unevaluated binding of symbol s.

7. Use `s=quote(s)` to clear symbol s.

8. Use `float(pi)` to get the floating point value of π. Set `pi=float(pi)` to evaluate expressions with a numerical value for π. Set `pi=quote(pi)` to make π symbolic again.

9. Use `e=exp(1)` to assign the natural number e to symbol `e`.

10. Assign strings to unit names so they are printed normally. For example, setting `meter="meter"` causes symbol `meter` to be printed as meter instead of m_{eter}.

11. Use `expsin` and `expcos` instead of `sin` and `cos`. Trigonometric simplifications occur automatically when exponentials are used. See also `expform` for converting an expression to exponential form.

12. Use `rect(expform(f))` to maybe find a new form of trigonometric expression `f`.

    ```
    f = cos(theta/2)^2
    rect(expform(f))
    ```

    $\frac{1}{2}\cos(\theta) + \frac{1}{2}$

13. Complex number functions `conj`, `mag`, etc. treat undefined symbols as representing real numbers. To define symbols that represent complex numbers, use separate symbols for the real and imaginary parts.

    ```
    z = x + i y
    conj(z) z
    ```

    $x^2 + y^2$

```
z = A exp(i theta)
conj(z) z
```

A^2

14. Use `mag` for component magnitude, `abs` for vector magnitude.

```
y = (a, -b)
mag(y)
```

$$\begin{bmatrix} a \\ b \end{bmatrix}$$

```
abs(y)
```

$[a^2 + b^2]^{1/2}$

15. Use `draw(y[floor(x)],x)` to plot the values of vector `y`.

```
y = (1,2,3,4)
draw(y[floor(x)],x)
```

16. The following exercise[1] demonstrates some `eval` tricks. Let

$$\psi = \frac{\phi_1 + \phi_2}{2} \exp\left(-\frac{iE_1 t}{\hbar}\right) + \frac{\phi_1 - \phi_2}{2} \exp\left(-\frac{iE_2 t}{\hbar}\right)$$

where ϕ_1 and ϕ_2 are orthogonal and operator A has the following eigenvalues.

$$\begin{aligned} A\phi_1 &= a_1\phi_1 \\ A\phi_2 &= a_2\phi_2 \end{aligned}$$

Verify that

$$\langle A \rangle = \int \psi^* A \psi \, dx = \frac{a_1 + a_2}{2} + \frac{a_1 - a_2}{2} \cos\left(\frac{(E_1 - E_2)t}{\hbar}\right)$$

Because ϕ_1 and ϕ_2 are normalized we have $\int \phi_1^* \phi_1 \, dx = 1$ and $\int \phi_2^* \phi_2 \, dx = 1$. By orthogonality we have $\int \phi_1^* \phi_2 \, dx = 0$. Hence the integral can be accomplished with `eval`.

```
phi1 = r1(x) exp(i theta1(x)) -- note that conj(phi1) phi1 == r1(x)^2
phi2 = r2(x) exp(i theta2(x)) -- note that conj(phi2) phi2 == r2(x)^2

psi = 1/2 (phi1 + phi2) exp(-i E1 t / hbar) +
      1/2 (phi1 - phi2) exp(-i E2 t / hbar)

A(f) = eval(f, phi1, a1 phi1, phi2, a2 phi2) -- eigenvalues

f = conj(psi) A(psi)

Abar = eval(f, r1(x)^2, 1, r2(x)^2, 1, r1(x) r2(x), 0) -- integral

Abar == (a1 + a2) / 2 + (a1 - a2) / 2 cos((E1 - E2) t / hbar)
```

[1] See exercise 4-10 of *Quantum Mechanics* by Richard Fitzpatrick.