Using *Mathematica*

If you want to learn Mathematica in depth, get the Mathematica book, or one of the other books available on the program. But the following tidbits of info should get you started.

☞ **Evaluating expressions:**

In the Windows or Mac versions of Mathematica, as well as the X-window front-end under Unix, to get Mathematica to evaluate an expression, type `<Shift><Enter>`. In other words, hold down the Shift key and hit Enter (or Return). This is equivalent (but easier) than doing 'Evaluate Cell' from the pull-down menus. If you are using the text-based front-end, then just hit `<Return>`

To get off the ground, try entering (and having Mathematica evaluate) `2+2`. Or `2^3`. Or `10!` Note that a caret ^ denotes exponentiation (raising something to a power).

☞ **Defining variables:**

To define numerical value for constants, use something like `M = 100`, or `z = 1.3 10^(-6)`. Use `Pi` for π, `E` for e, and `I` for $i \equiv \sqrt{-1}$. Try evaluating $e^{i\pi}$ by entering `E^(I Pi)`.

☞ **Vectors and matrices:**

To enter a vector, enclose the comma-separated list of components in braces {}, as in $v = \{1, 2, 3\}$. A matrix is entered as a list-of-lists, with each row enclosed in braces and commas between rows. For example, the matrix $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ may be entered as `m = {{a,b},{c,d}}`.

A single period (.) represents a dot product or matrix multiplication. Given the above definitions of v and m, try evaluating $v.v$, or $m.v$, or $m.m$.

☞ **Defining functions:**

To define a function of one variable, such as $f(x) = \sin x$, type `f[x_]:= Sin[x]`. (And if you are using the notebook front-end, type `<Shift><Enter>` to make Mathematica evaluate this definition.) Note that:

- ❖ All built-in functions have Capitalized Names.
- ❖ Brackets, not parentheses, are used to enclose function arguments.
- ❖ You must append an underscore to the symbolic name of the variable on the left-hand-side if you want this definition to apply for any choice of dummy variable [that is, if you want $f(y)$ to be treated as $\sin y$, and you must use :=, not just = (the colon means "delay evaluation until a number takes the place of y").
- ❖ To define a function of two variables, use something like `g[x_,y_] := Cos[x y]`.

☞ **Useful Built-in Functions:**

➢ **Expand**

Expands algebraic expressions. For example, `Expand[(a+b)^2]` returns $a^2+2ab+b^2$

➢ **Series**

Performs power series expansions of functions. For example,

```
Series[Sin[x],{x,0,5}]
```

expands $\sin x$ about $x = 0$ up through order x^5, returning $x-x^3/6+x^5/120+O(x^6)$.

- **Sum**

  Performs finite sums, and infinite sums which can be done exactly. For example,

  ```
  Sum[1/n^2,{n,1,Infinity}]
  ```

  will evaluate $\sum_1^\infty 1/n^2$ and returns $\pi^2/6$.

- **NSum**

  Performs (finite or infinite) sums numerically. For example,

  ```
  NSum[Log[n] Sin[n]/n^3,{n,1,Infinity}]
  ```

  returns 0.198126.

- **Integrate**

  Performs both definite and indefinite integrals. For example,

  ```
  Integrate[x^5 E^(-x),{x,0,Infinity}]
  ```

  evaluates $\int_0^\infty x^5 e^{-x}\, dx$ and returns 120; `Integrate[x E^(-x),x]` does the indefinite integral of $\int x e^{-x}\, dx$ and returns $e^{-x}(-1-x)$.

- **NIntegrate**

  Same syntax as `Integrate` but performs definite integrals numerically.

- **Solve**

  Solves equations (symbolically). May return multiple solutions. For example,

  ```
  Solve[a x^2 + b x + c == 0, x]
  ```

  returns a list containing the two substitutions,

- **DSolve**

  Solves differential equations (symbolically). See the Mathematica book, or online documentation, for details.

- **NDSolve**

  Solves differential equations numerically. See the book, or online documentation, for details.

- **Inverse**

  Gives the inverse of a square matrix.

- **Transpose**

  Takes the transpose of a matrix.

- **Plot**

  Plots one or more functions. For example,

  ```
  Plot[{x,Log[1+x]},{x,0,2}]
  ```

  plots both x and $\ln x$ for x from 0 to 2. Note that `Plot` accepts numerous options for customizing the appearance of a plot – see the book, or online docs, for details.

- **ContourPlot**

  Generates 2D contour plots of a function. Try, for example,

  ```
  ContourPlot[Sin[x y],{x,0,2Pi},{y,0,2Pi}].
  ```

Additional useful functions include `ComplexExpand`, `Simplify`, `Collect`, `Det`, `Eigenvalues`, and `Eigenvectors`. See the book, or on-line docs, for details.