Mathematica Tip Sheet

Built-In Constants:

π = `Pi` e = `E` $i = \sqrt{-1}$ = `I` ∞ = `Infinity`

Built-In Functions:

`Abs[x]`	`Sin[x]`	`ArcSin[x]`
`Sqrt[x]`	`Cos[x]`	`ArcCos[x]`
`Exp[x]`	`Tan[x]`	`ArcTan[x]`
`Log[x]` ($= \ln x$)	`Sec[x]`	`ArcSec[x]`
`Log[b, x]` ($= \log_b(x)$)	`Csc[x]`	`ArcCsc[x]`
`n!` or `Factorial[n]`	`Cot[x]`	`ArcCot[x]`

Grouping:

Parentheses - `( )`	Used for grouping for basic operations, like `+, -, *, /, ^`.
Square Brackets - `[ ]`	Used for functions to indicate the variable quantity to be used. (`f[x]`).
Curly Braces - { }	Used for lists, vectors, matrices, and ranges of values for options.

Assigning Values:

`x = value`	Assigns *value* to the variable `x`.
`x = y = value`	Assigns *value* (the same value) to <u>both</u> the variables `x` and `y`.
`Clear[x,y]`	Clears all values (if any) previously assigned to `x` and `y`. (USE OFTEN!)
`x == y`	Tests whether `x` is equal to `y`, often used when trying to solve equations.
expr`/.x-> value`	Replaces every `x` in *expr* with *value*.
expr`/.{x-> xval, y-> yval}`	Replaces `x` and `y` in *expr* with *xval* and *yval*, respectively.
`f[x_]= expr`	Defines a function `f`, of one variable. Remember the underscore (`_`)!
`g[x_, y_]= expr`	Defines a function `g`, of two variables.

Some Algebra Commands:

`Expand[`*expr*`]`	Multiplies out products and powers in the *expr*.
`Factor[`*expr*`]`	Factors *expr* over the integers.
`Apart[`*expr*`]`	Decomposes *expr* into partial fractions.
`Simplify[`*expr*`]`	Performs algebraic transformations to give the simplest form of *expr*.
`Solve[`*lhs*`==`*rhs*`,x]`	Solves the polynomial equation *lhs*`=`*rhs* (exactly) for `x`. (Notice the double equal sign `==`.)
`FindRoot[`*lhs*`==`*rhs*`,{x,a,b}]`	Numerically solves the polynomial equation *lhs*`=`*rhs* for `x`, starting in the interval `(a, b)`.
`a=x/.Solve[`*lhs*`==`*rhs*`,x]`	Stores the solution value as the variable `a`. If there is more than one solution, add `[[n]]` at the end of the command to store the n^{th} result as `a`.
`sol=x/.FindRoot[`*lhs*`==`*rhs*`,{x,a,b}]`	Stores the solution value as the variable `sol`.
`Solve[{`*eq1*`,`*eq2*`,...,`*eqN*`}, {x1,x2,...,xN}]`	Solves a system of `N` equations (written with `==`), for the variables `x1,...,xN`.

Manipulating Lists and Vectors:

`letters={a,b,c}`	A list called `letters` with three entries, `a`, `b`, and `c`. OR A vector called `letters` with components, `a`, `b`, and `c`.
`letters[[n]]`	Returns the nth element in the list called `letters`. (`letters[[3]] = c`).
`Dot[u,v]` or `u.v`	Returns the dot product of two vectors `u` and `v`.
`Cross[u,v]`	Returns the cross product of two <u>three-dimensional</u> vectors `u` and `v`.
`Table[f[x],{x,a,b,n}]`	Creates a table (list) of values of `f[x]`, going from `x=a` to `x=b` in increments of `n`. (If no increment is specified, the default value of 1 is used.)
`Tableform[list]`	Prints the elements of a list in a vertical table.

Some Calculus Commands:

`D[expr,x]`	Finds $\frac{d}{dx}(expr)$.
`D[expr,{x,n}]`	Finds $\frac{d^n}{dx^n}(expr)$.
`f'[x]`	Finds the first derivative of a previously defined function `f[x]`.
`f''[x]`	Finds the second derivative of a previously defined function `f[x]`.
`Integrate[expr,x]`	Evaluates the indefinite integral $\int expr\ dx$.
`Integrate[expr,{x,a,b}]`	Evaluates the definite integral $\int_a^b expr\ dx$.
`Limit[expr,x->a]`	Evaluates $\lim_{x \to a} expr$.
`Sum[a[n],{n,a,b}]`	Evaluates $\sum_{n=a}^{b} a[n]$.

Some Graphics Commands:

`Plot[f[x],{x,a,b},options]`	Creates a 2D plot of `y=f[x]` for the interval $a \le x \le b$.
`Plot[{f[x], g[x]},{x,a,b},options]`	Creates a 2D plot of `y=f[x]` and `y=g[x]` on a single set of axes.
`Plot3D[f[x,y],{x,a,b},{y,c,d},options]`	Creates a 3D plot of `z=f[x,y]` over the region $a \le x \le b$, $c \le y \le d$.
`ParametricPlot[f[t],{t,a,b},options]`	Creates a 2D plot of the parametrically defined function `f[t]={x[t],y[t]}` for $a \le t \le b$.
`ParametricPlot3D[f[t],{t,a,b},options]`	Creates a 3D plot of the parametrically defined function `f[t]={x[t],y[t],z[t]}` for $a \le t \le b$.
`ListPlot[{{x1,y1},{x2,y2},{x3,y3}}]`	Plots the points with coordinates $(x1, y1)$, $(x2, y2)$, $(x3, y3)$.
`Show[{graph1,graph2},options]`	Displays the two graphs *graph1, graph2* on a single set of axes.

Some Selected Plot Options:

`AspectRatio->value`	Sets the height-to-width ratio for the plot.
`Axes->False`	Exclude axes in the plot. (Default is `True`).
`AxesLabel->{xlabel,ylabel}`	Labels to put on the axes.
`PlotPoints->value`	The number of points to plot. (Default is 25).
`PlotRange->{min,max}`	The range of values to display on the plot.
`PlotStyle->{Thickness[w]}`	Gives all curves a thickness of `w` as a fraction of the plot width.
`PlotStyle->{RGBColor[a,b,c]}`	Produces color graphs: `a`, `b`, and `c` are values between 0 and 1 which represent the saturation of red, green, and blue, respectively.

A Few Other Useful Commands:

`%`	Refers to the last answer output from Mathematica. Caution: This is the last output generated, which is not necessarily the answer directly above the line on which `%` is entered.
`N[expr,n]`	Returns a decimal value for *expr*, with `n` significant digits.
`//N`	When typed after another command, converts it to a numerical (decimal) result.
Semicolon: `;`	Used at the end of successive lines of input, it evaluates, but suppresses output.
Space:	Used between two variables, it indicates a multiplication. For example, `x y` (with the space) means `x*y`, but `xy` (without any space) refers to a variable name.

Some Menu Commands / Other Items:

SHIFT + ENTER	Executes an input cell.
Cell → Delete All Output	Deletes all output in the active notebook.
Evaluation → Abort Evaluation	Stops the evaluation of a cell which is currently running.
Palettes → BasicMathInput	Opens general palette with basic symbols.
Evaluation → Quit Kernel → Local	Clears all memory in current session.
Help → Documentation Center	Opens the Help Browser.