

# Useful Commands in *Mathematica*

Dr. Leon Arriola  
Mathematical & Computer Sciences Department  
University of Wisconsin–Whitewater©2006

## 1 Basic Operations on Lists

A list is always enclosed by curly braces  $\{\text{obj}_1, \text{obj}_2, \dots\}$ . The following built-in commands are useful in manipulating lists:

Append[ ]	Element[ ]	LUdecomposition[ ]
AppendTo[ ]	* Extract[ ]	MatrixExp[ ]
* Array[ ]	* First[ ]	* MatrixForm[ ]
ColumnForm[ ]	** Flatten[ ]	MatrixPower[ ]
Complement[ ]	FlattenAt[ ]	MatrixQ[ ]
Count[ ]	Fold[ ]	** Max[ ]
Cross[ ]	FoldList[ ]	MemberQ[ ]
* Delete[ ]	FreeQ[ ]	** Min[ ]
DeleteCases[ ]	* IdentityMatrix[ ]	Minors[ ]
Depth[ ]	Inner[ ]	NullSpace[ ]
* Det[ ]	* Insert[ ]	Order[ ]
* DiagonalMatrix[ ]	* Intersection[ ]	OrderedQ[ ]
Dimensions[ ]	* Inverse[ ]	Outer[ ]
* Dot[ ]	* Join[ ]	* PadLeft[ ]
* Drop[ ]	JordanDecomposition[ ]	* PadRight[ ]
** Eigensystem[ ]	* Last[ ]	* Part[ ]
** Eigenvalues[ ]	ListConvolve[ ]	** Partition[ ]
** Eigenvectors[ ]	ListCorrelate[ ]	PartitionsP[ ]

PartitionsQ[ ]	SchurDecomposition[ ]	VectorQ[ ]
* Permutations[ ]	Select[ ]	
** Position[ ]	Signature[ ]	
* Prepend[ ]	SingularValues[ ]	
* PrependTo[ ]	* Sort[ ]	
PseudoInverse[ ]	Split[ ]	
QRDecomposition[ ]	** Table[ ]	
Range[ ]	TableAlignments[ ]	
* Replace[ ]	TableDepth[ ]	
* ReplaceAll[ ]	TableDirections[ ]	
ReplaceList[ ]	* TableForm[ ]	
* ReplacePart[ ]	TableHeadings[ ]	
** ReplaceRepeated[ ]	TableSpacing[ ]	
Rest[ ]	* Take[ ]	
Reverse[ ]	TensorRank[ ]	
RotateLeft[ ]	Tr[ ]	
RotateRight[ ]	* Transpose[ ]	
RowReduce[ ]	* Union[ ]	

## 2 Basic Graphics Commands

### 2.1 2 Dimensional Plots of Functions

The basic 2 dimensional plot format is

$$\text{Plot}[f[x], \{x, x_{\min}, x_{\max}\}]$$

where  $f$  is the function to be plotted, with argument  $x$  in the interval  $x_{\min} \leq x \leq x_{\max}$ . The following options are available:

* AbsoluteDashing	Compiled	* GridLines
* AbsolutePointSize	DefaultColor	GrayLevel
** AspectRatio	DisplayFunction	Hue
* Axes	* Epilog	ImageSize
* AxesLabel	FormatType	MaxBend
* AxesOrigin	* Frame	PlotDivision
* AxesStyle	* FrameLabel	** PlotLabel
Background	* FrameStyle	** PlotPoints
ColorOutput	* FrameTicks	* PlotRange

PlotRegion  
 \* PlotStyle  
 \* PointSize  
 \* Prolog  
 RGBColor  
 RotateLabel  
 TextStyle  
 Thickness  
 \* Ticks

## 2.2 2 Dimensional Plots of Points

The format for plotting a list of points is

$$\text{ListPlot}[\{y_1, y_2, \dots, y_n\}]$$

$$\text{ListPlot}[\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}]$$

The following options are available:

AspectRatio	Epilog	PlotLabel
Axes	Frame	PlotRange
AxesLabel	FrameLabel	PlotRegion
AxesOrigin	FrameStyle	PlotStyle
AxesStyle	FrameTicks	Prolog
Background	GridLines	RotateLabel
ColorOutput	ImageSize	Ticks
DefaultColor	PlotJoined	

## 2.3 2 Dimensional Parametric Plots

The command for plotting the parametric curves  $x = x(t)$  and  $y = y(t)$  where  $t_{\min} \leq t \leq t_{\max}$  is

$$\text{ParametricPlot}[\{x[t], y[t]\}, \{t, t_{\min}, t_{\max}\}]$$

$$\text{ParametricPlot}[\{\{x_1[t], y_1[t]\}, \{x_2[t], y_2[t]\}, \dots, \{x_n[t], y_n[t]\}\}, \{t, t_{\min}, t_{\max}\}]$$

The options that are available are the same as those for Plot[].

## 2.4 Contour Plots

The command for plotting 2 dimensional contour plots of  $f = f(x, y) = c$ , where  $c$  is a constant, and where  $x_{\min} \leq x \leq x_{\max}$  and  $y_{\min} \leq y \leq y_{\max}$  is

$$\text{ContourPlot}[f[x, y], \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$$

The options that are available are

AspectRatio	ContourLines	FrameTicks
Axes	Contours	ImageSize
AxesLabel	ContourShading	PlotLabel
AxesOrigin	ContourSmoothing	PlotPoints
AxesStyle	ContourStyle	PlotRange
Background	DefaultColor	PlotRegion
ColorFunction	Epilog	Prolog
ColorFunctionScaling	Frame	RotateLabel
ColorOutput	FrameLabel	Ticks
Compiled	FrameStyle	

## 2.5 3 Dimensional Plots of Functions

The command for plotting the function  $f = f(x, y)$ , where  $x_{\min} \leq x \leq x_{\max}$  and  $y_{\min} \leq y \leq y_{\max}$  is

$$\text{Plot3D}[f[x, y], \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$$

The following options are available:

AmbientLight	Compiled	PlotRegion
AspectRatio	DefaultColor	Prolog
Axes	Epilog	Shading
AxesEdge	FaceGrids	SphericalRegion
AxesLabel	HiddenSurface	Ticks
AxesStyle	ImageSize	ViewCenter
Background	Lighting	ViewPoint
Boxed	LightSources	ViewVertical
BoxRatios	Mesh	
BoxStyle	MeshStyle	
ClipFill	Plot3Matrix	
ColorFunction	PlotLabel	
ColorFunctionScaling	PlotPoints	
ColorOutput	PlotRange	

## 2.6 3 Dimensional Parametric Plots

The command for plotting 3 dimensional parametric plots  $x = x(t)$ ,  $y = y(t)$ , and  $z = z(t)$ , where  $t_{\min} \leq t \leq t_{\max}$ , or  $x = x(s, t)$ ,  $y = y(s, t)$ , and  $z = z(s, t)$ , where  $s_{\min} \leq s \leq s_{\max}$  and  $t_{\min} \leq t \leq t_{\max}$

$$\text{ParametricPlot3D}[\{x[t], y[t], z[t]\}, \{t, t_{\min}, t_{\max}\}]$$

$$\text{ParametricPlot3D}[\{x[s, t], y[s, t], z[s, t]\}, \{s, s_{\min}, s_{\max}\}, \{t, t_{\min}, t_{\max}\}]$$

The following options are available:

AmbientLight	Compiled	PlotRegion
AspectRatio	DefaultColor	PolygonIntersections
Axes	Epilog	Prolog
AxesEdge	FaceGrids	RenderAll
AxesLabel	ImageSize	Shading
AxesStyle	Lighting	SphericalRegion
Background	LightSources	Ticks
Boxed	Plot3Matrix	ViewCenter
BoxRatios	PlotLabel	ViewPoint
BoxStyle	PlotPoints	ViewVertical
ColorOutput	PlotRange	

## 3 Assignments

### 3.1 Set & SetDelayed

Immediate Assignment: `Set:`  $\text{lhs} = \text{rhs}$ , e.g., `f[x_] = x^2`

Delayed Assignment: `SetDelayed:`  $\text{lhs} := \text{rhs}$  e.g., `f[x_] := x^2`

Unset: `f =.`

`Clear[f]`

## 4 Programming

### 4.1 Logical Operations

`Equal[x, y]` or  $x == y$

`Unequal[x, y]` or  $x != y$  or  $x \neq y$

`Less[x, y]` or  $x < y$

`Greater[x, y]` or  $x > y$

`LessEqual[x, y]` or  $x \leq y$

`GreaterEqual[x, y]` or  $x \geq y$

`x == y == z` all equal  
`x != y != z` all not equal, i.e., distinct  
`x < y < z` all increasing  
`x > y > z` all decreasing

## 4.2 Loops & Conditionals

For the following, multiple lines of expression are separated by the semicolon ;

`Do[expression, k]` Evaluates the expression  $k$  times.

`Do[expression, i, imax]` Evaluates the expression  $imax$  times, starting with  $i = 1$ , and incrementing  $i$  by 1.

`Do[expression, i, imin, imax]` Evaluates the expression starting at  $i = imin$  and ending at  $i = imax$  and incrementing  $i$  by 1.

`Do[expression, i, imin, imax, increment]` Evaluates the expression as previous, in increments of `increment`.

`Do[expression, i, imin, imax, j, jmin, jmax]` Nested Do loop. Evaluates the expression where  $i$  changes by 1 for each cycle of  $j$ .

`Do[expression, i, imin, imax, i_increment, j, jmin, jmax, j_increment]` Same as previous with specified increments for  $i, j$ .

`While[condition, expression]` Evaluates the condition, then evaluates expression until the condition is `False`.

`For[initialization, test, increment, expression]` Executes the initialization, then repeatedly evaluates the expression, increments until `test` is `False`.

`If[condition, true, false]` Evaluates condition and executes the true expression if condition is `True` and executes the false expression if condition is `False`.

`If[condition, true]` Evaluates condition and executes `true` if condition is `True` and takes no action when condition is `False`, and returns `Null`.

`If[condition, false]` Evaluates condition and executes `false` if condition is `False` and takes no action if condition is `True` and returns `Null`.

`If[condition, true, false, neither]` Evaluates condition and executes `true` if condition is `True`, executes `false` if condition is `False` and executes `neither` if condition is neither `True` or `False`.

## 4.3 Decrement & Increment

`Decrement[x]` or `x--`

`Increment[x]` or `x++`

`PreDecrement[x]` or `--x`

`PreIncrement[x]` or `++x`

## 5 Operations on Strings

`StringDrop[ ]`

`StringForm[ ]`

`StringInsert[ ]`

`StringJoin[ ]`

`StringLength[ ]`

`StringMatchQ[ ]`

`StringPosition[ ]`

`StringReplace[ ]`

`StringReplacePart[ ]`

`StringReverse[ ]`

`StringSkeleton[ ]`

`StringTake[ ]`

`StringToStream[ ]`