Wolfram Mathematica CHEAT SHEET

BASIC PRINCIPLES

Evaluate input	⇧ ↵		
Abort calculation	Alt ,	⌘ ,	
Quit	Alt F4	⌘ Q	`Quit[ ]`

SPECIAL CHARACTERS

ⅇ	Esc ee Esc	2.71828...
ⅈ	Esc `ii` Esc	√-1 (also `I`)
π	Esc p Esc	3.14159...
→	Esc `->` Esc	replace
×	Esc * Esc	times
×	Esc cross Esc	cross prod
√	Esc sqrt Esc	square root

NOTATION

`;`	Suppress output	`[]`	Arguments
`/.`	`Replace`	`{}`	List
`//.`	`Replace` repeatedly	`()`	Precedence
`=`	assign	`->`	rule
`:=`	assign delayed	`:>`	delayed rule
`/@`	`Map` (see below)	`/;`	`Condition` (see below)
`@@`	`Apply` (see below)	`(* *)`	comment

SELECTED OPERATORS

=	Assignment	=
`==`	Equality	Esc == Esc
≡	`SameQ`	Esc === Esc
≠	Inequality	Esc != Esc
∧	And	Esc && Esc
∨	Or	Esc || Esc
∪	`Union`	Esc un Esc
∩	Intersection	Esc inter Esc
*	Conjugate	Esc co Esc
`x=.`	clear value of x	

COMMON TOOLS & CONSTRUCTS

`N[`*expr*`]`	`N[`*expr*`,` *prec*`]`	*expr*`` ` ``*prec*	Yields numerical value of *expr* to precision *prec* (also postfix `//N`)
`Evaluate`		`Hold`	Forces or prevents evaluation (useful for `Table` or `Replace` in `Plot`, for instance)
`Condition[`*expr*`,`*cond*`]`		*expr* `/;` *cond*	Creates a pattern matching only if *cond* is `True`

PROCEDURAL PROGRAMMING

`Do[`*expr*`,{`*i*`,`i_{min}`,`i_{max}`}]` `While[`*test*`,`*body*`]` `For[`*start*`,`*test*`,`*incr*`,`*body*`]` `If[`*cond*`,`*iftrue*`,`*iffalse*`]`

`Table[`*expr*`,{`*i*`,`i_{min}`,`i_{max}`}]`	Generates vectors, matrices, tensors, and other arrays from *expr* across the range of *i*
`Module[{`*x*`,`*y*`,...},`*expr*`]`	Specifies that occurrences of symbols *x*, *y* are local to *expr*
`With[{`*x*`=`x_0`,`*y*`=`y_0`,...},`*expr*`]`	Specifies that each occurrence of symbols *x*, *y* should be replaced by x_0,y_0 in *expr*

FUNCTIONAL PROGRAMMING

`Replace[`*expr*`,`*rules*`]`	`/.`	`//.`	Applies *rules* to transform *expr*; rules are of the form `{rule1->expr1, ...}`
`Map[`*f*`,`*expr*`]`	`/@`		Applies *f* to each element of *expr*; particularly useful with lists
`Apply[`*f*`,`*expr*`]`	`@@`	`@@@`	Replaces the head of *expr* by *f*; particularly useful with lists (`@@@` for first level)
`Thread[`*f*`[`*args*`]]`			Threads *f* over any lists appearing in *args*; see also `MapThread`
`Select[`*list*`,`*cond*`]`			Chooses elements of *list* for which *cond* is `True`
expr`[`*#*`] &`			Creates a pure function applying *expr* to *#*

ALGEBRA, TRIGONOMETRY, & SERIES

Functions are typically named intuitively, and it is straightforward to find what you want once you know it exists.

`Sin[`*x*`]`	sin(x)	$\sum_{i=1}^{\infty} \frac{7}{10^i}$	Esc sum Esc Ctrl +_ `i=1` Ctrl +% Esc inf Esc → Ctrl +/ 7 ↓ 10 Ctrl +^ i
`Sinh[`*x*`]`	sinh(x)		
`Log[`*x*`]`	$\log_e(x) = \ln(x)$	`Fit[`*data*`,`*fns*`,`*vars*`]`	Finds a least-squares fit to a list of *data* to functions *fns*
`Exp[`*x*`]`	exp(x)	`Interpolation[`*data*`]`	Creates a pure function of order `InterpolationOrder`.
`Expand[ ]`		`Maximize[ ]`	
`NSolve[ ]`		`Minimize[ ]`	

CALCULUS & DIFFERENTIAL EQUATIONS

FUNCTIONS		**OPTIONS (EXAMPLES, NOT RECOMMENDATIONS)**	
D[*expr*,*x*] *expr*'	Give the partial derivative of *expr* w.r.t. *x*	NonConstants→{u_i}	Specifies u_i implicitly depend on *x*
Integrate[*expr*,*x*]	Gives the indefinite integral of *expr*		
$\int_0^\infty f(t)\, e^{-st}\, dt$	Esc int Esc Ctrl +_ 0 Ctrl +% Esc inf Esc → f[t] Esc ee Esc Ctrl +6 -s t → 10 Esc dd Esc t		
NIntegrate[*f*,{*x*,x_{min},x_{max}},{*y*,y_{min},y_{max}},*opt*→*val*]		Method→"*MonteCarlo*"	PrecisionGoal→6
Dsolve[{*eqn*},*y*,*x*]		Analytically solves the differential equation (if possible)	
NDSolve[{*eqn*},*y*,{*x*,x_{min},x_{max}}]		Numerically solves the differential equation	
Plot[Evaluate[*y*[*x*] /. %], {*x*,x_{min},x_{max}}]			

TECHNICAL DATA

Execute data functions without arguments to determine scope; execute with string argument "Properties" *to find queryable data.*

ElementData	GenomeData	GraphData	Get <<*pkg*`	Introduce a package with new symbols and rules
IsotopeData	ProteinData	PolyhedronData	Needs	Required for some packages not commonly used
ChemicalData	GenomeLookup	Quantity[*magnitude*,*units*]		Represents a quantity of *magnitude units*
ParticleData	WeatherData	Import[*file*]		Load *file* (including XLS, HDF, PDB) or URL
AstronomicalData	GeodesyData	Export["*file*",*expr*,"*format*"]		Export data of *expr* to *file* as *format*

MANIPULATION & DYNAMIC CONTENT

Dynamic content allows you to explore large parameter spaces with vastly abbreviated output relative to Table.

Dynamic[*expr*]	Returns an object representing the current value of *expr*; try with Plot, for instance
Animate[*expr*,{*t*,t_{min},t_{max}}]	Generates animation of *expr* by varying *t* continuously from t_{min} to t_{max}
Manipulate[*expr*,{*t*,t_{min},t_{max},*dt*}]	Generates a version of *expr* with controls allowing manipulation of *t* in steps *dt*

PLOTTING

SELECTED FUNCTIONS	**SELECTED OPTIONS**	
Plot[*f*,{*x*,x_{min},x_{max}},*opt*→*val*] Plot[{*f*,*g*},{*x*,x_{min},x_{max}},*opt*→*val*]	PlotRange→ {{x_{min},x_{max}},{y_{min},y_{max}}}	PlotStyle→{{Orange, Thickness[0.003],Dotted}}
ListPlot[{{x_1,y_1},{x_2,y_2},...}]	Axes→True	PlotLabel→"Curves"
ContourPlot[*f*,{*x*,x_{min},x_{max}},{*y*,y_{min},y_{max}},*opt*→*val*]	AxesOrigin→{*x*,*y*}	Frame→True
PolarPlot[*r*,{*θ*,θ_{min},θ_{max}},*opt*→*val*]	AxesLabel→Automatic	AspectRatio→1/5
Plot3D[*f*,{*x*,x_{min},x_{max}},{*y*,y_{min},y_{max}},*opt*→*val*]	AxesStyle→Thick	ColorFunction→"Rainbow"
ParametricPlot[{f_x,f_y},{*t*,t_{min},t_{max}},*opt*→*val*]	Ticks→{{0,π,2π,3π},{0,1}}	Mesh→300
Show[$grfx_1$,$grfx_2$,...,*opt*→*val*]	Legended[*data*,*style*]	RotateLabel→True