Cheatsheet

Properties of Tian Xiao
tinyurl.com/tx-ma1102r-sagemath

* represents an **optional** argument.
Italics represents **name** of an argument.

Basic Command

=	Assign a value
==	Equal sign
reset(**vars*)	Delete and reset vars
show(*expr*)	Change display mode
_	Quote previous input
.n(*digits = *value*)	Approximations
var(**var*)	Define a variable
assume(*assumption*)	Make an assumption
forget(**assumption*)	Forget an assumption
assumptions()	Show all assumptions
.full_symplify()	Simplify an expression
.factor()	Factorise an expression
.expand()	Expand an expression

Basic Algebra

+-*/^	Basic operators
.sqrt()	Square root
^(1/*n*)	Principle n-th root
.nth_root(*n*)	Real n-th root
pi	π
.sin()	Trigonometric functions
.log(**base*)	Logarithm functions
e	Euler constant
.exp()	Exponential functions
.factorial()	Factorial

Function and Equation

f(x) = *expr*	Assign a function
.substitute(x = *value*) | f(*value*)	Substitute a value into a function
(*min*, *max*) | [*min*, *max*]	Open/closed interval
piecewise([[*interval*, *f(x)*], ...])	Piecewise functions
plot(*function*, (**var*, **xmin*, **xmax*), *ymin = *value*, *ymax = *value*, *plot_points = *num*, *color = *color*, *detect_poles = "show")	Plot a graph. To show the asymptote, key in the last kwarg
*equation.find_root(*xmin*, *xmax*)	Find roots in a certain interval
implicit_plot(*equation*, (*xmin*, *xmax*), (*ymin*, *ymax*))	Implicit plot
parametric_plot((*x(t)*, *y(t)*), (t, *tmin*, *tmax*))	Parametric plot
.solve(*x*, *algorithm = "*sympy*")	Solve an equation

Differentiation

.limit(x = *value*, *dir = "-")	Calculate the limit
Infinity | oo	Infinity
.derivative(x, **order*)	Differentiate with respect to x
.rhs()	Right-hand side
.implicit_derivative(y, x, **order*)	Implicit differentiate dy/dx

Integration

.sum(*var*, *min*, *max*)	Sum an expression
.integral(x, *(x, *xmin*, *xmax*))	Integrate with respect to x

Differential Equation

desolve(*ODE*, y(x), *ics = [x0, y0, **moreics*], *contrib_ode = True)	Solve a DE. If cannot be solved, we can try the last kwarg
_C | _K1 | _K2	Constants in DE
desolve_rk4(*ODE*, y(x), ics = [x0, y0, *moreics], *end_points = [*xmin*, *xmax*], *step = *d*, *output = "plot")	Approximate the solution of a DE
list_plot()	Plot a list