(customised) SymPy Cheatsheet (https://sympy.org)

Basics

Sympy help:	`help(function)`
Declare symbols:	`var('x y z')`
Declare real symbol:	`x = Symbol('x', real=True)`
Substitution:	`expr.subs(old, new)`
Numerical evaluation:	`expr.n(); N(expr)`
Expanding:	`expand(expr)`
Simplify expression:	`simplify(expr)`
Previous answer(s):	`_ ; _3`

Constants

π:	`pi`
e:	`E`
∞:	`oo`
i:	`I`

Number types

Integers ($\mathbb{Z}$):	`Integer(x)`
Rationals ($\mathbb{Q}$):	`Rational(p, q)`
Sympify $\frac{2}{3}$:	`S(2)/3`
Reals ($\mathbb{R}$):	`Float(x)`

Simplification and rewriting

Factoring:	`factor() factorint()`
Rational expr.:	`ratsimp() cancel() collect()`
Partial fractions:	`apart()`
Powers:	`powsimp() powdenest()`
Expand powers:	`expand_power_exp()`
Expand powers:	`expand_power_base()`
Square roots:	`radsimp() sqrtdenest()`
Trigonometric:	`trigsimp() expand_trig()`
Logarithm:	`expand_log() logcombine()`
Force application:	`...(..., force=True)`
Rewrite using 'func':	`expr.rewrite(func)`
Combinatorial:	`combsimp()`
Find exact repres.:	`nsimplify(0.333333)`

Basic functions

Trigonometric:	`sin cos tan cot`
Cyclometric:	`asin acos atan acot`
Hyperbolic:	`sinh cosh tanh coth`
Area hyperbolic:	`asinh acosh atanh acoth`
Exponential:	`exp(x)`
Square root:	`sqrt(x)`
Logarithm ($\log_b a$):	`log(a, b)`
Natural logarithm:	`log(a)`
Absolute value:	`abs(x)`

Equations

Equation $f(x) = 0$:	`solve(f, x)`
Newer alternative:	`solveset(f, x)`
Solve in $\mathbb{R}$:	`solveset_real(f, x)`
System of equations:	`solve([f, g], [x, y])`
Differential equation:	`dsolve(equation, f(x))`

Calculus

$\lim_{x \to a_+} f(x)$:	`limit(f, x, a)`
$\lim_{x \to a} f(x)$:	`limit(f, x, a, dir='+-')`
$\lim_{x \to a_-} f(x)$:	`limit(f, x, a, dir='-')`
$\frac{d}{dx} f(x)$:	`diff(f, x)`
$\frac{\partial}{\partial x} f(x, y)$:	`diff(f, x)`
$\int f(x)\,dx$:	`integrate(f, x)`
$\int_a^b f(x)\,dx$:	`integrate(f, (x, a, b))`
Taylor series (at a, deg n)	`series(f, x, a, n)`
Formal power series	`fps(f, x, a)`
Res(f, z_0)	`residue(f, z, z0)`

Discrete math

Factorial ($n!$):	`factorial(n)`
Binomial coefficient $\binom{n}{k}$:	`binomial(n, k)`
Sum ($\sum_{n=a}^{b} expr$):	`summation(expr, (n, a, b))`
Product ($\prod_{n=a}^{b} expr$):	`product(expr, (n, a, b))`

Unevaluated "noun" forms

Sum, product:	`Sum(expr,(n,a,b)) ; Product()`
Limit:	`Limit(f, x, a)`
Derivative, integral:	`Derivative() ; Integral()`
Evaluate:	`Integral(f, x).doit()`

Complex numbers

Re(), Im():	`re() im()`
Expand:	`expand_complex()`
Re, Im as tuple:	`expr.as_real_imag()`
Arg(z), $\lvert z \rvert$:	`arg(z) Abs(z)`
$\bar{z}$:	`conjugate(z)`

Linear algebra

Matrix definition:	`m = Matrix([[a, b], [c, d]])`
Column vector:	`c = Matrix([a, b])`
Row vector:	`r = Matrix([[a, b]])`
Transpose:	`r = c.T`
Determinant:	`m.det()`
Inverse:	`m.inv()`
Identity $n \times n$:	`eye(n)`
Zeros, Ones $n \times n$:	`zeros(n) ones(n)`

Plotting (Jupyter notebook)

Interactive plots:	`%matplotlib notebook`
Basic plot:	`plot(f)`
Custom limits:	`plot(f,(x,a,b),ylim=(ya,yb))`
Multiple functions:	`plot(f, g, h, (x,a,b))`
Combine plots:	`plot1.extend(plot2)`
Parametric plot:	`plot_parametric()`
Implicit plot:	`plot_implicit(expr)`
3D surface plot:	`plot3d(f(x,y))`
3D line plot:	`plot3d_parametric_line()`
3D param. surf.:	`plot3d_parametric_surface()`

Printing

mathjax:	`init_printing(use_latex='mathjax')`
LaTeX print:	`print latex()`
Pretty print:	`display()`