

Index

!=, 362
", 6, 11
@interact, 423
#, 55
\$, 159
%, 220
' , 6, 11
(, 1
) , 1
*, 1
+, 1
-, 1
., 2
/, 1
//, 220
:, 319
<, 362
<=, 362
==, 362
>, 362
>=, 362
?, 70
[, 14
\, 32
], 14
^, 2
_, 3
|-->, 101
 π , *see* pi
 ω , 310

abs, 22, 313
absolute values, 22, 361
abstract algebra, 68–70
acceleration function, 120, 121
acos, 17
acosh, 226
acot, 17
acoth, 226
acsc, 17
acsch, 226
add_constraint, 267
addition, 1
air resistance, 125
amicable pairs, 217–218
and, 405
append, 382
arbitrary lists, *see* lists, arbitrary
arccos, 16
arccosh, 226
arccot, 17
arccoth, 226
arccsc, 17
arccsch, 226
arcsec, 17
arcsech, 226
arcsin, 16
arcsinh, 226
arctan, 17
arctanh, 226
arrays, 381–387
arrow, 158
as_sum_of_permutations, 246
asec, 17
asech, 226
asin, 17
asinh, 226
aspect_ratio, 178
assert, 402
AssertionError, 403
assertions, 402–403
assume, 81
asymptotes, 27–29
atan, 17

- `atan2`, 19
- `atanh`, 226
- `augment`, 204
- averaging numbers, 404–406
- `axes_labels`, 156

- balancing chemical equations, 115–120
- `bin`, 235
- binary, 234–235
- `binomial`, 224
- biology, 109–112
- Boolean variables
 - see variables, Boolean, 272
- braces, 4
- brackets, 3
- `break`, 372

- capitalization, 11
- Cardano's cubic formula, 61–62
- `CC`, 403
- `ceil`, 224
- chaos theory, 281, 330–331
- `charpoly`, 244
- chemical reactions, 115–120
- chemistry, 115–120, 230–232
- `cholesky`, 246
- coinage, 332–334
- combinations, 224
- commands
 - misspelled, 435
 - parameters to, 435–436
- commas, 2, 13, 440
- comments, 55, 348–349
- comparitors, 362
- compilers, 414
- complex numbers, 14–15, 60–61, 305–315
 - conjugates of, 306
 - functions of, 307
 - imaginary part, 306
 - magnitude of, 312
 - norms of, 307
 - real part, 306
- composition of functions, *see* functions,
 - composition of
- confidence intervals, 396–400
- `conjugate`, 306
- constants, 336
- constraints, 267–269
- `continue`, 414
- `continued_fraction`, 264
- continued fractions, 264–266
- `contour_plot`, 173

- contour plots, *see* plots, contour
- control flow, 354–370
- convergents, 265
- converting degrees and radians, 17
- convex functions, *see* functions, convex
- copy-and-paste, 436
- `cos`, 16
- cosecant, 16
- cosine, 16
- `cot`, 16
- cotangent, 16
- Coulomb's Law, 185, 187
- counter-battery fire, 124–125
- `cross_product`, 212
- cryptology, 127–135, 400–402
- `csc`, 16
- cubic equations, 61
- curl of a vector field, 296–299

- debugging, 414
- decimals, 7–8, 47
- `def`, 332
- defining functions, *see* functions,
 - defining
- demand curve, 156
- `derivative`, 74
- derivatives, 74–76
 - partial, 200–201, 257, 290
 - plotting, 75
- `desolve`, 274
- `det`, 211
- determinants, *see* matrices,
 - determinants
- dice
 - rolling, 395–396
- dictionaries, 413
- `diff`, 74
- differential equations, 273–285
- discrete randomness, 394–396
- discriminant, 308
- divergence, 294–295
- `divides`, 375
- divisibility, 375–377
- division, 1
- divisors, 216–219
- `divisors`, 52
- docstrings, 70, 332
- `dot_product`, 212

- `e`, 8
- early termination, 359–362
- `echelon_form`, 206
- eigenvalues, *see* matrices, eigenvalues

- eigenvalues, 244
- eigenvectors, *see* matrices, eigenvectors
- eigenvectors_left, 244
- eigenvectors_right, 244
- Eisenstein Integer, 311
- elif, 368
- else, 366
- environmental science, 136–141
- erf, 88
- error handling, 409–410
- estimation error, 396–400
- euler_phi, 223
- Euler's Phi function, 222–223
- evaluate, 1, 7
- except, 409
- exceptions, 363–365
- exp, 12
- expand, 51
- exponential functions, *see* functions, exponential
- exponents, 2

- factor, 51
- factorial, 130, 224
- factorials, 237–238, 370–371
- factoring, 51–52
- feasible points, 267
- Fermat's Last Theorem, 311
- Fermat's Little Theorem, 130
- find_fit, 232
- find_root, 63
- floor, 224
- for, 319
- for loops, *see* loops, for
- frobenius, 246
- full_simplify, 89, 262, 309, 313
- function, 275
- functions
 - composition of, 48–49
 - convex, 256
 - declaring, 275
 - defining, 45–46
 - exponential, 12
 - hyperbolic trig, 226
 - inverse trig, 16–17
 - multivariate, 50, 193–194
 - plotting, *see* plots, functions
 - reciprocal trig, 16
 - referring to, 99–102
 - velocity, 121, 122

- gcd, 52
- generating tables, 319–320

- geometric series, 262–263
- get_values, 267
- global variables, *see* variables, global
- gradients, 184–185, 188–189, 201, 259
- gram_schmidt, 246
- graphs, *see* plots
- greatest common denominator, 213–215
- greatest common divisor, 52–53
- gridlines, 29–30

- half-evaluation, 194
- help, 70–74, 441
- hermite_form, 246
- hessenberg_form, 246
- hex, 235
- hexadecimal, 234–235
- higher-order roots, 15–16
- huge numbers, 237–240
- hyperbolic trig functions, *see* functions, hyperbolic trig

- identity_matrix, 203
- if, 354
- if statements, 354, 362
- if-then-else statements, 365–367
- imag_part, 306
- implicit_plot, 170
- in, 319, 384
- indentation, 320, 437
- industrial optimization, 112–115
- inequalities
 - systems of, 266–273
- infeasible points, 267
- infinite series, 260–264
- infinite sums, 260–264
- infinitely many solutions, 43–45
- initial-value problems, 276–277
- int, 235
- integer linear programs, *see* linear programming, integer
- integral, 77
- integrals, 76–93, 160
 - definite, 78
 - Gaussian error function, 88–89
 - impossible, 82–84
 - improper, 79–81
 - indefinite, 77
 - multiple, 299–300
- integrate, 81
- integration
 - by partial fractions, 78
 - numerical, 86–88
- interacts

- building, 417–432
- integral, 431–432
- tangent line, 420–428
- web templates for, 426
- interest
 - compound, 2, 13, 65–66, 195–199, 320
 - simple, 1
- internet connection, 434–435
- inverse, 206
- `inverse_laplace`, 287
- inverse trig functions, *see* functions,
 - inverse trig
- `is_prime`, 213
- iteration, 346–348
- `jordan_form`, 246
- Lagrange multipliers, 256–260
- `lambda`, 414
- `laplace`, 286
- Laplace Transforms, 285–288
- Laplacians, 291–292, 294
- L^AT_EX, 159, 240–241, 288, 300–305
- `latex`, 241
- law of cosines, 20–21
- `lcm`, 214
- least common multiple, 213–215
- least squares, *see* regression, least squares
- `left_kernel`, 209
- `left_nullity`, 209
- lemniscates, 168
- `len`, 217
- Leontief Analysis, 141–154
- `lhs`, 408
- `limit`, 227
- limits, 227–228, 324–326
- line breaks, 436–437
- linear programming, 266–273
 - environmental applications, 136–141
 - integer, 272
 - nutritional applications, 272–273
 - shipping applications, 112–115
- linear systems, 35–36, 41, 56–57, 247–250
- `list`, 225
- list notation, 54
- lists, 381–387
 - arbitrary, 322
 - of points, 381–383
 - operations on, 383–385
- local variables, *see* variables, local
- log, 12
- log-log plots, *see* plots, log-log
- logarithms, 12–14, 60
- logical operators, 412
- loops, 319–331, 337–354
 - accumulators, 322–323
 - `for`, 319–331
 - wasteful, 323
 - `while`, 370–381
- LU, 246
- macroeconomics, 141–154
- magnitude
 - of a complex number, 312
 - of a vector, 212
- mapping, 100
- matrices, 33–45, 201–211, 241–250
 - coordinates, 38–39
 - determinants, 211, 242, 290–291
 - eigenvalues, 241–245, 256
 - eigenvectors, 241–245
 - exponentiation, 203
 - factorizations, 246–247
 - Hessian, 256, 289–291
 - inverses, 206–208
 - Jacobian, 292–293
 - kernels of, 208–211
 - multiplication, 202–203
 - reduced row echelon form, 34, 36–40, 206
 - Vandermonde, 41–43
- `matrix`, 37
- `max`, 224, 407
- maximizations, 267–269
- maximizing profit, 104
- maximizing revenue, 104
- `mean`, 407
- `median`, 407
- microeconomics, 104–108, 179–180
- `min`, 224, 407
- minimizations, 254–260
- `minimize`, 255
- minimizing costs, 104
- `minpoly`, 245
- `MixedIntegerLinearProgram`, 268
- `mod`, 221
- modular arithmetic, 220–222
- Monte Carlo Method, 387–402
- mortgages, 195–199
- multiplication, 1
- multivariable formulas, 56, 193–194
- multivariate functions, *see* functions, multivariate

- N, 7
- `new_variable`, 271
- Newton's formula for gravity, 199–200, 252–254
- Newton's method, 339–354, 357–362
 - for loop, 342–343
 - tabular, 341
 - while loop, 372–374
- `next_prime`, 213
- non-linear systems, 57–60
- `norm`, 186
- `nth_prime`, 215
- number theory, 213–223
- `numerical_approx`, 8
- numerical approximation, 8, 345–346
- `numerical_integral`, 88
- nuts
 - mixed, 272–273
- object-oriented programming, 414
- `oct`, 235
- octal, 234–235
- omega, 310
- optimization
 - constrained, 256–257
 - unconstrained, 255–256
- `or`, 404
- parabolas, 21
- parallel tangent lines, 352–353
- parameters, 283–285, 346–348
 - ordering of optional, 356–357
- `parametric_plot`, 181
- parentheses, 1, 3
 - mismatched, 14, 435
- `partial_fraction`, 78
- Permutations, 225
- permutations, 224–226
- personal finance, 195–199
- physics, 185–188
- pi
 - built-in, 8
 - estimating, 388–389
 - historical context, 48, 266
- `pi`, 8
- `plot`, 21
- `plot_loglog`, 190
- `plot_slope_field`, 279
- `plot_vector_field`, 182
- `plot3d`, 184
- plots
 - 2D, 21–33
 - adding arrows, 158
 - adding text, 158–159
 - contour, 173–180, 188–189
 - dashed lines, 161
 - dotted lines, 161
 - functions, 46–47
 - grids, 156–158
 - hyperactive functions, 163–164
 - implicit functions, 170–173
 - integrals, 160
 - labeling axes, 156
 - log-log, 189–191
 - multiple graphs, 29–32, 75
 - parametric 2D, 180–182
 - polar, 164–168
 - restricted domains, 161–162
 - saving, 23
 - scatter, 228–232
 - solution curves, 282–283
 - streamline, 283
 - vector, 278
 - vector field, 135, 182–189, 299
- `point`, 29
- points of agreement, 351–352
- Poiseuille's Law, 109–112
- polar coordinates, 164–168, 312–313
- `polar_plot`, 165
- polyhedra, 271
- polynomials, 50–53
 - expanded form, 51
 - factored form, 51
 - Taylor, 251–254, 328–330
- precision control, 9
- present value of an annuity, 4
- price-demand curve, 41
- prime numbers, 213–216, 371–372
- `prime_range`, 213
- `print`, 4
- `prod`, 384
- pseudorandom numbers, 400–402
- QQ, 403
- QR, 246
- quartic equations, 240
- quick search, 73
- quotes, 29
- `raise`, 363
- `random`, 389
- `range`, 22
- `range`, 319
- `rational_form`, 246
- `real_nth_root`, 15, 312
- `real_part`, 306

- reciprocal trig functions, *see* functions,
 - reciprocal trig
- recursion, 413
- regression, 137, 231–234
 - least squares, 247–250
- remainders, 219–220
- remove, 384
- `remove_constraint`, 270
- repeated roots, *see* roots, repeated
- resources
 - programming, 412
- return, 348
- returning only certain roots, *see* roots,
 - returning only certain
- `rhs`, 408
- `right_kernel`, 209
- `right_nullity`, 209
- roots, 52, 60, 62–70
 - complex, 307–308
 - cube, 309–310
 - repeated, 374
 - returning only certain, 407–409
 - solving for multiple, 66–67
- rounding, 224
- rounding error, 326–328
- RR, 403
- `rref`, 37
- `RuntimeError`, 365

- SageMathCloud, 443–445
- SageTeX, 301–305
 - `\sage`, 302–303
 - `\sageplot`, 303
 - `\sagestr`, 304–305
 - `sagesilent`, 304
- saturation point, 105
- saving a plot, *see* plots, saving
- `scatter_plot`, 190, 229
- scatter plots, *see* plots, scatter
- scientific notation, 10–11
- scope, 413
- `search_doc`, 72
- `sec`, 16
- secant, 16
- `selector`, 430
- selectors, 430–431
- `set_objective`, 267
- sharing your results, 93–99
 - QR code reader, 95
 - save as, 95
 - save as PDF, 98
- shipping costs, 113
- `show`, 30, 38, 61

- `shuffle`, 385
- `sigma`, 216
- significant digits, 8
- simulations, 414
- `sin`, 16
- single-variable formulas, 54–55
- `slider`, 425
- sliders, 423–425
- slope fields, 277–283
- `smith_form`, 246
- smooth numbers, 129–130
- `solve`, 54
- `solve_left`, 203
- `solve_right`, 203
- source code, 71
 - displaying, 71
- speed of sage, 236–237
- `sqrt`, 7
- square roots, 7–8, 14–15
- squirrels, 171–172
- `str`, 305
- strings, 6, 413
 - raw, 159
- subroutines, 331–339, 420–425
- subtraction, 1
- Sudoku, 236
- `sudoku`, 236
- `sum`, 176, 260, 384
- summation identities, 261–262
- summation proofs, 263–264
- `switch`, 413
- `symplectic_form`, 246
- syntax, 433

- tab completion, 72
- `tan`, 16
- `taylor`, 251
- Taylor polynomials, *see* polynomials,
 - Taylor
- `text`, 158
- thermodynamics, 175–178
- `timeit`, 236
- timeouts, 437–438
- torpedo problem, 283–285
- `transpose`, 248
- trial division, 374–380
- trigonometry, 16–21
 - hyperbolic, 226
- `try`, 409

- `var`, 55
- variable data types, 412
- variables, 47–48

- Boolean, 272
 - declaring, 55, 433–434
 - global, 336
 - local, 336
- vector**, 186
- vector calculus, 288–300
- vector field plots, *see* plots, vector field
- vector notation, 289
- vectors, 211–212
 - cross products, 211–212
 - dot products, 211–212
 - length, 211–212
- velocity functions, *see* functions,
 - velocity
- verbosity control, 354–356
- viewing windows, 24–26
- volume flow rate, 109

- wasteful loops
 - see* loops, wasteful, 323
- weak_popov_form**, 246
- web templates, *see* interacts, web
 - templates for
- while**, 370
- while** loops, *see* loops, **while**

- zigzag_form**, 246
- ZZ**, 378, 403