

INDEX

Symbols

+ (addition operator), 2
{ } (curly brackets), to define
 a set, 122
 δ (delta), 184
/ (division operator), 2
 ε (epsilon), 192, 197–199
== (equality operator), 124
 e (Euler’s number), 179
** (exponential operator), 3
// (floor division operator), 2–3
j (imaginary number, in Python), 6
 \int (integral), 200
 \cap (intersection), 127
 λ (lambda), 192, 197–199
% (modulo operator), 3
* (multiplication operator), 2
 π (pi), 129, 147
- (subtraction operator), 2
 θ (theta), 49
→ (transformation), 158
 \cup (union), 126

A

abs() function, 7
acos() function, 179
algebraic expressions. *See*
 expressions
Anaconda software installer,
 213–214
animation
 of a growing circle, 153–155
 of a projectile’s trajectory,
 156–158
animation module, 154
Anscombe, Francis, “Graphs in
 Statistical Analysis,” 82

Anscombe’s quartet, 82–83
antiderivative, 200
append() method, 30
area
 of a circle, estimating, 145–146
 between two curves, 206–207
asin() function, 179
aspect ratio, 153
atan() function, 179
ATM example, 138–140
attractors, 172

B

bar charts
 for exercise, 57–59
 for expenses, 56–57
Barnsley fern, drawing, 163–168
break, exiting with, 24

C

calculus. *See* functions
cardinality, of a set, 122
cards, shuffling deck of, 144–145
Cartesian coordinates, 28–29
 graph, 32
Cartesian product, 127–128, 137
causation, and correlation, 76
circles
 animating, 153–155
 drawing, 151–153
 estimating area of, 145–146
 packing a square with, 168–169
close() method, 231
cmath module, 7
code, reusing, 235–236
coin tosses, 137–138, 144
comma-separated value (CSV)
 files, 86–88

- complex numbers, 6–7
 - addition and subtraction of, 6
 - cmath module, 7
 - complex() function, 6
 - complex roots, 22
 - conjugate of, 7
 - conjugate() function, 7
 - magnitude, 7
 - Mandelbrot set, 172–176
 - real and imag, 7
- components() function, 226
- continuous compound interest, 183–184
- controlling program exit, 24
- correlation coefficients, 75–78, 89
- cos() function, 52, 178–179
- Counter class, 66
- CSV (comma-separated value) files, 86–88
- csv module, 86
 - next() function, 87
 - reader() function, 87
- curves
 - area between, 205–206
 - length of, 207–208

D

- data. *See also* sets, statistical measures
 - dispersion, measuring, 71–75
 - reading from files, 83–88
- deck of cards, shuffling, 144–145
- definite integral, 200–201, 203
- derivative, of a function, 185–191
 - calculator program, 186–187
 - higher-order, 188–191
 - partial, 187
- Derivative class, 185, 189
- dictionary, 100–101, 224–226, 227
- die rolls
 - game, 135–136
 - law of large numbers, 143–144
 - simulating, 134–135
 - target score, possibility of, 136–137

- discrete probability, 131–139
- dispersion of data, measuring, 71–75
 - range, finding, 71–72
 - variance and standard deviation, finding, 72–75
- distribution, uniform, 131
- domain, of a function, 178
- Droettboom, Michael, “matplotlib,” 150

E

- else block, 230
- empty lists, 30
- enumerate() function, 31
- epsilon (ϵ), 192, 197–199
- equations, solving, 20–22, 105–108
 - graphically, 115
 - linear, 20, 108
 - quadratic, 20–22, 106
 - solve() function, 105–108, 180, 199
 - for variables, 106–107
- Euler’s number (e), 179
- even-odd vending machine, 22–23
- events (probability), 131
- exception handling, 9, 228–235
 - file reading errors, 232–235
 - multiple exception types, 228–229
 - try...except, 9, 228
 - try...except...else, 230
 - ValueError, 9, 12
 - ZeroDivisionError, 11, 228–229
- exit option, for programs, 24–26
- exp() function, 204
- expenses, visualizing with bar charts, 56–57
- experiments (probability), 131
- expressions, 96–105
 - factorizing and expanding, 96–97
 - multiplying, 104–105
 - plotting, 108–115
 - input by the user, 111–113
 - multiple, 113–115

- pretty printing, 97–100
- strings, converting to, 103–105
- substituting in values, 100–103

extrema, of a function, 188–191

F

factor() function, 96–97, 115

factors of an integer,

- calculating, 12–14

fargs keyword argument, 154, 158

Fibonacci sequence, 59–60

file handling

- close() method, 231
- filename as input, 232
- handling errors, 232–235
- open() function, 231
- reading files, 230–231
- readlines() method, 232

file object, 84

formatting output, 15

- format(), 15
- number of digits, 16
- print() function, 1

fractals, 158–168

- Barnsley fern, 163–168
- Hénon’s function, 171–172
- Mandelbrot set, 172–176
- Sierpiński triangle, 170–171
- transformations of points, 158–163

fractions

- calculator, 23–24
- working with, 5–6

fractions module, 5

frames argument, 154, 158

frequency tables, creating, 69–71

FuncAnimation class, 154–158

functions (calculus), 178

- common, 178–180
- continuity at a point,
 - verifying, 205
- derivatives of, finding, 185–187
 - higher-order, 188–191
- domain of, 178
- extrema of, 188–191

- integrals of, finding, 200–201
- limit of, finding, 181–185
- probability density, 201–204
- range of, 178

G

geometric shapes, drawing,

- 150–158

geometric transformations, 158

global maxima and minima,

- 188–199

golden ratio, 59–60

gradient ascent method, 191, 195

gradient descent method, 199,

- 205–206

graphs, creating with matplotlib,

- 32–46
 - customizing with titles and labels, 41–44
 - marking points, 33–35
 - saving as images, 45–46
 - temperature data example, 35–44

H

higher-order derivatives of

- functions, finding, 188–191

Hénon’s function, 171–172

Hunter, John, “matplotlib,” 150

I

IDLE, 1, 13–14

- new program, 13
- program execution, 14
- running a program, 14
- shell, 1

importing, modules, 5

imshow() function, 172

indefinite integral, 200

index, of a list, 29, 31

inequalities, solving, 117–119

infinite loop, 24

Infinity, 183, 204

- in operator, 122
- input() function, 8
- installation, of software
 - on Linux, 216–217
 - on Mac OS X, 217–220
 - on Windows, 214–215
- Integral class, 200
- integrals of functions, finding, 200
- intersection, of sets, 127
- interval argument, 154

K

- keys, in a dictionary, 224, 227

L

- labels, 4
- Lady ferns, 164
- law of large numbers, 144
- legend() function, 40
- len() function, 62
- limit, finding, 181
- Limit class, 182
- Linux, software installation on,
 - 216–217
- lists, 29–31
 - appending to a list, 30
 - choosing a random element, 161
 - creating a set, 123
 - empty lists, 30
 - index, 29
 - iterating over the elements, 31
 - len() function, 62
 - list comprehensions, 223–224
 - lists of lists, 173–175
 - max() function, 72
 - min() function, 72
 - sort() method, 64
 - sum() function, 62
 - tuples as members, 66
 - zip() function, 77
- local maxima and minima,
 - 188–191
- log() function, 179

M

- Mac OS X, software installation on,
 - 217–220
- Mandelbrot set, 172–176
- mathematical operations, 1–3
 - exponential operator, 3
 - floor division operator, 2
 - modulo (%) operator, 3, 12
- math module, 178
- matplotlib, 32
 - animation module, 154
 - axes
 - auto scaling, 152
 - customizing, 42
 - Axes object, 151
 - axis() function, 43
 - barh() function, 57
 - Circle patch, 151
 - colorbar() function, 175
 - displaying images, 172
 - documentation, 211
 - Figure object, 150, 154
 - FuncAnimation class, 154–158
 - gca() function, 152
 - gcf() function, 154
 - imshow() function, 172
 - labels, 41
 - legend, adding a, 40
 - legend() function, 40
 - marker, 34
 - multiple data sets, 38, 53
 - patches, 150
 - plot() function, 32, 36
 - Polygon patch, 168
 - pylab module, 32
 - pyplot module, 44
 - savefig() function, 45
 - saving, 45–46
 - scatter() function, 81
 - scatter plots, 79, 81–83
 - set_aspect() method, 153
 - show() function, 32
 - title, 41
 - title() function, 41
 - xlabel() function, 41
 - ylabel() function, 41

- maxima and minima, of functions, 188–191
- max() function, 72
- mean, finding, 62–63
- median, finding, 63–65
- min() function, 72
- mode, finding, 65–69
- modules, 5
- modulo (%) operator, 3
- multiplication tables, generating, 15–17, 23
- multiplying expressions, 104–105

N

- __name__, 221–223
- negative index, of a list, 31
- NegativeInfinity, 204
- Newton’s law of universal gravitation, 46–48
- number line, 28
- numbers
 - abs() function, 7
 - common number sets, 126
 - complex numbers. *See* complex numbers
 - conversion between types, 5
 - float() function, 5
 - floating point, 4–5
 - Fraction class, 5, 6
 - fractions module, 5
 - integers, 4–5
 - int() function, 5
 - is_integer() method, 10
 - random. *See* random numbers
 - rational, irrational, and real, 126
 - type() function, 4
 - types of, 4–7
- Nykamp, Duane Q., “The idea of a probability density function,” 202

O

- open() function, 231
- order of operations (PEMDAS), 3

P

- Packages (Python), 32
- partial derivative of functions, finding, 187
- Pearson correlation coefficient, 75
- PEMDAS (order of operations), 3
- pi (π), estimating value of, 147
- plot() function, 32, 109
- plotting
 - expressions, 108–115
 - input by the user, 111–113
 - multiple, 113–115
 - with formulas, 46–54
 - projectile motion, 48–54
 - using SymPy. *See* SymPy
 - polynomial expressions, 117
 - polynomial() method, 119
 - pretty printing, 97–100
 - probability, 131–140, 201–204
 - continuous random variable, 201
 - density functions, 201–204
 - distribution, uniform, 131
 - expectation, 143
 - law of large numbers, 144
 - nonuniform probability, 164
 - random numbers. *See also* random numbers
 - generating, 134–137
 - nonuniform, 137–140
 - random variable, 143
 - Project Euler, 210
 - projectile motion, 48, 191
 - animation, 156
 - trajectory drawing, 51, 56
 - pylab module, 32
 - pyplot module, 44–45
 - Python
 - documentation, 210, 211
 - IDLE, 1, 13–14
 - installation
 - Linux, 216–217
 - Mac OS X, 217–220
 - Windows, 214–215
 - overview, 221–236

Q

- quadratic equations
 - finding the roots of, 20–22
 - solving, 106
- quadratic functions, exploring
 - visually, 55–56

R

- random module, 134
 - choice() function, 160
 - randint() function, 134, 175
 - random() function, 134
 - uniform() function, 146
- random numbers
 - ATM example, 138–140
 - coin tosses, 137–138, 144
 - deck of cards, shuffling, 144–145
 - die rolls. *See* die rolls
 - generating, 134–137
 - nonuniform, 137–140
- range
 - of a function, 178
 - of a set, 71–72
- range() function, 13, 37, 50
 - start, stop, and step values, 13
- rate of change, finding, 184
- reading data from files, 83–88
 - CSV files, 86–88
 - text files, 84–85
- return values, multiple, 226–228
- reusing code, 235–236
- Robertson, Ian, “Calculating Percentiles,” 90

S

- sample spaces (probability), 131
- save() function, 111
- saving plots, as image files,
 - 45–46, 111
- scatter plots, 79, 81–83
- series
 - calculating value of, 102–103
 - Fibonacci, 59–60
 - printing, 99–100
 - summing, 116
- set_aspect() method, 153

- sets, 121–131
 - cardinality, 122
 - checking for a number in, 122
 - common, 126
 - correlation between, 75–81
 - creating, 122–124
 - empty, 123
 - from lists or tuples, 123
 - EmptySet object, 123
 - FiniteSet class, 122
 - FiniteSet object, 122
 - intersect() method, 127
 - is_subset() method, 124
 - is_superset() method, 124
 - iterating through the
 - members, 123
 - operations, 126–131
 - Cartesian product, 127–128
 - formulas, using sets of
 - variables in, 129
 - gravity example, 130–131
 - union and intersection, 126
 - powerset() method, 125
 - repetition and order, 123–124
 - subsets, supersets, and power sets, 124–125
 - union() method, 126–127
 - Venn diagrams, 140–143
- show() function, 32, 111
- shuffling, deck of cards, 144–145
- Sierpiński triangle, 170–171
- simultaneous equations, 108
- sin() function, 52, 178, 179
- software installation
 - on Linux, 216–217
 - on Mac OS X, 217–220
 - on Windows, 214–215
- solving algebraic equations, 105
- standard deviation, finding, 72–75
- statistical measures
 - correlation coefficient,
 - 75–81, 87
 - calculating, 76–78
 - high school grades example,
 - 78–81
 - dispersion, 71–75
 - frequency tables, 69–71
 - grouped, 90–91

- mean, 62–63
- median, 63–65
- mode, 65–71
- Pearson correlation
 - coefficient, 75
- percentile, 89–90
- range, 71–72
- standard deviation, 72–75
- variance, 72–75
- step size, 192, 197–199
- string, 8
 - format() method, 15
 - int() and float(), See under numbers, 8
- strings to mathematical expressions, 103
- sum() function, 62
- summing a series, 116
- symbolic math, 93
- SymPy
 - as_numer_denom() method, 118
 - assumptions, 180
 - Derivative class, 185
 - documentation, 98, 211
 - doit() method, 182, 185
 - expand() function, 96
 - expression, factorizing an, 96
 - factor() function, 96
 - init_printing() function, 98
 - installation. *See* installation, of software
 - Integral class, 200
 - is_polynomial() method, 119
 - is_rational_function()
 - method, 119
 - Limit class, 182
 - plot() function, 109
 - plotting expressions with, 108–115
 - input by the user, 111–113
 - multiple, 113–115
 - Poly class, 117
 - pprint() function, 97–100
 - pretty printing, 97–100
 - save() function, 111
 - show() function, 111
 - simplify() function, 101
 - solve() function, 105, 106, 180
 - solve_poly_inequality()
 - function, 117
 - solve_univariate_inequality()
 - function, 118
 - solving inequalities, 117
 - S class, 182
 - subs() method, 100, 108, 184
 - summation() function, 116
 - symbol, defining a, 94
 - Symbol class, 94
 - symbols() function, 95
 - SympifyError class, 104
 - sympify() function, 103, 119, 186

T

- tan() function, 179
- title() function, 41–42
- trajectory (projectile motion)
 - comparing, 53–54, 56
 - drawing, 51–53
- transformation of a point, 158
- tuples, 29–31
 - empty, 31
 - iterating through the members, 123

U

- union, of sets, 118, 126–127
- units of measurement, converting, 17–20, 23
- universal gravitation, Newton’s law, 46–48
- user input
 - complex() function, 12
 - fractional numbers, 11
 - getting, 8–12
 - handling invalid input, 9–11
 - input() function, 8

V

- ValueError, 9, 12
- variables, 4, 178
 - nonlinear relationship, 47
- variance, finding, 72–75
- Venn diagrams, 140–143

W

while loop, 24
 exiting early using break, 24
Windows, software installation on,
 214–215

Z

ZeroDivisionError, 11, 228–229
zip() function, 77