

Index

- `*args/**kwargs`, 461–465
- `__init__`, *see* objects, constructors
- 2.7.x, Python, 6, 20, 197, 277, 427
- algorithms, 139, 153, 597, 601, *see also* performance, algorithms
- Anaconda
 - distribution, 6–7, 40
 - Navigator, 6, 17, 40
- animation, *see* Matplotlib, animation
- AppVeyor, 700
- arguments, *see* parameters
- arrays, 99–103, 522, *see also* NumPy
 - array syntax, 101–103, 129, 153, 164, 190–191, 376, 379, 670, 678
 - booleans, 406–407
 - compatibility, input arrays, 201
 - n*-dimensional, 372–374
 - performance, *see* arrays, performance; performance
 - readability, 202, 522
 - result, assignment of, 288, 308
- attributes, 269–271
- booleans, 404–408, 413, 428, 488–490, *see also* types, booleans
- creating, 99, 128–132, 184–186
- data types, 129, 131, 271–272, 274–277, 296, 406
- dimensions, degenerate, 384
- elements
 - address, 99
 - assigning, multiple, 100, 189
 - index, 99, 137, 186, 246, 370, 384
 - index, negative, 99, 189, 205, 384
 - number of, *see* length; NumPy, `size`
- functions acting on, 132–134, 190–191, 372–374
- functions with axis control, 372–374
- functions, contrast with methods, 271
- index offsetting, *see* slicing, arrays
- length, 101, 131
- lists, contrast with, 101, 277–278
- looping, *see* looping
- masked, 492–493, *see also* ma
- memory, 129, 374, 379, *see also* memory
- memory
- methods, 271–272
- methods, contrast with functions, 271
- missing values, *see* arrays, masked;
- data analysis, missing values; ma
- n*-dimensional, 365–380, 579
 - columns, 368
 - reshaping, 374–376
 - rows, 369
 - shape, 368–370
 - sheets, 369
- one-dimensional, 124–126, *see also* arrays
 - shape, 184, 368
- performance, 414–415, 672
- reshaping, *see* arrays, *n*-dimensional, reshaping
- sheets, 366, *see also* arrays, *n*-dimensional, sheets
- slicing, *see* slicing
- two-dimensional, 176–182, 186–187
 - columns, 183, 187
 - rows, 183, 187
 - shape, 183–184, 368
- type, 103
- typecodes, *see* arrays, data types
- `assert`, 688
- assignment, 13–15, 49, 286–288
- asymptotic algorithmic complexity, *see* performance, asymptotic algorithmic complexity
- attributes, *see* objects; objects, attributes
- autotools, 699
- Azure, 700
- big- \mathcal{O} notation, *see* performance, asymptotic algorithmic complexity
- binary files, 578, 579
- binary numbers, 422
- binary search, *see* searching, binary
- booleans, *see* types, booleans
- branching, 139–153, 490, *see also* expressions, booleans; `if`; `if-elif`; `if-else`; operators, comparison; truth table; types, booleans; variables, booleans
- conditions, 140, 143, 223, 239–241
- nested, 230–232
- `break`, 233–235
- C/C++, 704–705
- calculations, *see* expressions; operators; variables
- `callable`, 557
- cartopy, 440–445, 453–456, *see also* Matplotlib
 - borders, 454
 - `central_lon`, 454
 - features, 454
 - formatting, 454
 - `GeoAxesSubplot`, 454
 - `LambertConformal`, 473
 - `LatitudeFormatter`, 455
 - `LongitudeFormatter`, 455
 - `PlateCarree`, 454
- `class`, 546
- classes, *see* objects
- `cmake`, 699
- color systems, *see* image processing
- commenting, 66–69

730 Index

- compiled languages, *see* C/C++;
 - cython; gcc; Fortran; numba;
 - performance
- conda, *see* packaging, conda
- continuation character, *see* line continuation
- `continue`, 233–235
- contour plots, *see* Matplotlib, contour plots
- coordinates, *see* Matplotlib, coordinates
- copy
 - `copy`, 289
 - `deepcopy`, 289
- copying, 286–290
- cProfile, *see* testing, cProfile
- csv, 297
- ctypes, *see* C/C++
- cython, 680, 702

- data analysis, 124–128, 148–153, 176–182, 408–414, *see also*
 - arrays; NumPy; pandas
 - automating, 332, 335
 - average, nearest-neighbor, 176–179, 246
 - missing values, 303, 483–494
 - processing with booleans, 488–490
- data structures, 511, 521–522, *see also*
 - arrays; dictionaries; lists;
 - pandas; queues; sets; stacks;
 - types
- debugging, 74, 156–158, 344, *see also* testing
 - Jupyter, *see* Jupyter, debugging
 - Spyder, *see* Spyder, debugging
- decorators, 675, 680, 688
- `def`, 33, 120, 520
- `del`, 30, 460
- `delattr`, 557
- derivative, 537
- dictionaries, 459–465, 509, 516–519, 602, 611
 - delimiters, 460
 - elements
 - delete, 460
 - deleting, 518
 - keys, 460, 526
 - referencing, 460, 518
 - types, 460
 - values, 460
 - iterable, 517
 - `keys`, 516, 518
 - order, 460, 517, 518
 - `values`, 516
- differential equation, 537–540
- `dir`, 273
- directories, 328–332, *see also* glob; os; recursion; shutil
 - copying, 338–340
 - leaf, 336, 636
 - listing contents, 340–341, 347
 - making, 335–336
 - moving, 337
 - path, 332, 634
 - absolute, 334
 - relative, 333
 - removing, 336
 - renaming, 337
 - separator character, 332
 - tree, 332, 339, 340, 633
 - working, current, 327, 333, 334, 341, 345, 346
- directory
 - tree, 633
- docstrings, *see* documenting, docstrings; NumPy, docstring format
- documenting, *see also* commenting
 - docstrings, 153–154, 659–660
 - how-to guides, 657
 - maintaining, 658
 - reference guides, 657
 - Sphinx, *see* Sphinx
 - tools, strength and weaknesses of, 658
 - topic guides, 657
 - tutorials, 657
- dynamically typed, *see* types, dynamic

- `elif`, *see* `if-elif`
- `else`, 143, 148, 245
- `else`, *see also* `if-elif`; `if-else`
- environment, *see* interpreter; Jupyter; Spyder
- epoch, 414
- Excel files, 571, 576–577, 579, *see also*
 - `ExcelWriter`; `openpyxl`;
 - pandas
 - multiple worksheets, 585
 - reading, 577
 - writing, 577
- `ExcelWriter`, 585
- exceptions, 298–300
 - class, 299, 300
 - custom, 562
 - handling, 300, *see also* `try/except`
 - multiple, 315
 - `raise`, *see* `raise`
 - `try/except`, *see* `try/except`
- expressions, 9–13, *see also* operators; types; variables
 - booleans, 222–230
 - negative sign, 44
 - order of evaluation, 11
 - parentheses, 9, 21
 - scientific notation, 10

- f2py, *see* Fortran
- file input/output, 261–267, 290–300, 669, 679
 - binary, 578
 - `close`, 291
 - comma-separated values, 296–298, *see also* csv
 - Excel files, *see* Excel files
 - file modes, 291
 - file opening errors, *see* exceptions
 - files of mostly numbers, reading, 296
 - `genfromtxt`, *see* NumPy, `genfromtxt`
 - netCDF files, *see* netCDF files
 - newline, 264
 - `open`, *see* `open`
 - pickle files, *see* pickle files
 - `readline`, 291
 - `readlines`, 291
 - text files, 261–262, 290–300
 - `with/as`, *see* `with/as`
 - `write`, 291
 - `writelines`, 292
- files, 328–331, 578, 592, *see also* binary
 - files; directories; file input/output; glob; os; text files
 - copying, 338–340
 - kinds, testing, 341–342
 - metadata, 338, 571, 579
 - names, 637
 - removing, 339

- float, 104, 293, 294
- fnmatch, 637
- folders, *see* directories
- for, *see* looping
- Fortran, 702–704
- functions, 27–29, 268, 560
 - calling, 29, 32, 34, 49, 643–644
 - defining, 32–35
 - naming, 33
 - parameters, *see* parameters
 - return value, 29, 34, 75, 153, 679
- gcc, 704
- generators, 634–637, 678–680
- getattr, 557
- Git, *see* version control, Git
- GitHub, 695
- GitLab, 695
- glob
 - glob, 341, 508, 600
- globals, 559
- grouping, *see* pandas, grouping
- hasattr, 557
- help, 32, 153, 272
- if, 140–143, 145, 232, *see also*
 - branching
- if-elif, 146–148, 231, 244, *see also*
 - branching
- if-else, 143–145, 232, *see also*
 - branching
- image processing, 394–398, 401–404, 422–423, *see also* Matplotlib
 - alpha, 404
 - channel intensities, 422–423
 - displaying an image, 403
 - grays, 423
 - imshow, 404
 - pixels, 401
 - reading an image, 403
 - RGB, 401–402
 - RGBA, 404
 - transparency, 404
 - writing an image, 403–404
- import, *see* importing
- importing, 30–32, 35, 45, 54, 124
- indentation, 33, 135, 143, 145, 150, 232, 238, 246
- inheritance, *see* objects, inheritance
- instances, *see* objects
- int, 104, 294
- interactive development environment, *see* Spyder
- interpreter, 9, 15–18, *see also* Jupyter; Spyder; terminal
- isinstance, 552
- Jupyter, 9, 17–18, 22, 37–38
 - debugging, 362
 - line magic, 55, 671, 699
 - Markdown, 66–69
- just-in-time compilation, 680–681
- lambda functions, 520, 529
- lazy evaluation, 679
- len, 59, 65, 612
- line continuation, 66, 70
- line plots, *see* Matplotlib, line plots
- line-profiler, *see* testing, line-profiler
- linting, *see* testing, linting
- list, 62, 612
- list comprehensions, 519–520, 584, 669
- lists, 56, 58–62, 309, 522
 - append, 278, 513, 639
 - arrays, contrast with, 101, 277–278
 - attributes, 278
 - concatenate, 639
 - count, 599
 - delimiters, 58
 - elements, 58, 59
 - address, 58
 - index, 58, 526
 - index, negative, 60
 - referencing, 59
 - types, 277, 467
 - index, 598
 - insert, 279
 - length, 59, 101
 - list comprehensions, *see* list comprehensions
 - looping, *see* looping
 - membership testing, *see* searching, membership testing
 - methods, 278
 - order, 58
 - pop, 279, 513
 - remove, 518
 - reverse, 303
- slicing, *see* slicing
- sort, 601
- types, 417, 418
- locals, 559
- looping, 134–139, 153, 378–380, 490, 522, 637, 640, 675
 - arrays, large, 138
 - arrays, raveling and, 375
 - arrays, through, 134–139, 148–153, 161
 - break, *see* break
 - continue, *see* continue
 - for, 135–139, 191, 232, 235, 631
 - infinite, 233, 235
 - iterables, 135–139, 235, 338, 515, 517
 - iterators, 135–139, 235
 - list comprehension, *see* list comprehension
 - nested, 191–193, 237–239, 251, 378–380
 - objects, *see* objects, looping through
 - performance, 379–380, 414–415, 668
 - strings, 136
 - while, 232–236, 642
- ma, 487, *see also* arrays, masked
 - filled, 493
 - fill_value, 493
 - get_fill_value, 493
 - masked_greater, 496
 - masked_greater_equal, 497
 - masked_less, 496
 - masked_less_equal, 497
 - masked_values, 492
 - set_fill_value, 493
- make, 661, 699
- map projections, *see* cartopy
- masked arrays, *see* arrays, masked
- Matplotlib
 - add_axes, 447, 466
 - add_feature, 454
 - animation, 440–445, 456–459
 - annotation, 95
 - API, object, 446–451
 - argument, third positional, 97
 - aspect ratio, 450, 467
 - Axes, 448
 - axes, 447, 448
 - AxesImage, 449, 456

732 Index

Matplotlib (Cont.)

`axis`, 93, 448, 456
`cartopy`, *see* `cartopy`
`cla`, 448, 456
`clabel`, 453, 469
`cmap`, 452
`color`, 94
color codes, 95
`colorbar`, 453
colormaps, 443, 450, 452, 469, 471
`colors`, 453
`contour`, 448, 451
contour line colors, 453
contour plots, 440–445, 451–453
`contourf`, 449, 451
convention, axis naming, 52
coordinates
 axis, 451
 data, 96, 451, 455
 normalized, 447, 466
curves, multiple, 97
`dpi`, 97, 98
`draw`, 456, 457
examples gallery, 84
`figsize`, 97, 467
`Figure`, 447
`figure`, 96, 97, 236, 447, 467
figure, current, 55, 96
figure, size, 97
figures, multiple, 96
font, size, 95
formatting
 argument, third positional, 94
 line, 93–94
 marker, 93–94
`gca`, 448
`GeoAxesSubplot`, 454
`get_cmap`, 452
importing, 54, *see also* importing
`imread`, 398, 403, 422
`imsave`, 403
`imshow`, 403, 449, 450
`ioff`, 458
`ion`, 456, 457
`label`, 97
`legend`, 97
legend, adding, 97
line plots, 52–54, 88–89
`Line2D`, 449, 456, 457
`LinearSegmentedColormap`, 450

`linestyle`, 94
linestyle codes, 94
`linewidths`, 453
map projections, *see* `cartopy`
`marker`, 94
marker codes, 95
`markerfacecolor`, 94
missing values, 491, *see also* arrays,
 masked; data analysis, missing
 values; pandas
`number`, 447
`orientation`, 453
`pause`, 456
`plot`, 52, 92, 96, 97, 448, 449
`projection`, 454
`pyplot`, 55, 84, 92, 450
`QuadContourSet`, 453
ranges, 93
rendering, inline, 55
resolution, 98
`savefig`, 98, 404
saving to a file, 98
`scatter`, 52, 55, 94, 449
scatter plots, 52–54, 94
`set_cmap`, 450
`set_data`, 450, 456
`set_extent`, 455
`set_linestyle`, 449
`set_major_formatter`, 455
`set_marker`, 449
`set_xdata`, 449, 456
`set_xticks`, 449, 455
`set_ydata`, 449, 456
`set_yticks`, 449
`show`, 55, 98
`size`, 96
`subplot`, 236
`subplots`, 447
subplots, multiple, 236–237
`text`, 95
ticks, 93
`title`, 95
`transform`, 451, 455
`XAxis`, 448
`xaxis`, 448, 455
`xlabel`, 62
`xticks`, 93
`YAxis`, 448
`yaxis`, 448, 455
`ylabel`, 62
`yticks`, 93

membership testing, *see* searching,
 membership testing
memory, 239, 272, 274–277, 422–423,
 668, 670, 676–678
memory-profiler, *see* testing,
 memory-profiler
methods, *see* objects; objects, methods
missing values, *see* data analysis,
 missing values
modeling, 209, 217, 537–544
 logistic equation, 540
 molecular dynamics, 262
 random walk, 209–217, 328–331
 timestep, 264
modules, 6, 30–32, 34–35, 559, 560
NaN, 483, 490–492
netCDF files, 573, 574–583
 dimensions, 580
 filling variable objects, 581
 global attributes, 580, 582
 reading, 582–583
 variable attributes, 580, 581
newline character, *see* strings, newline
None, 316, 541
numba, 680–681, 702
NumPy, *see also* arrays
 `allclose`, 310, 600
 `arange`, 129–130, 186, 207
 `array`, 99
 `astype`, 272, 294
 `average`, 491, 505
 `copy`, 289
 `cos`, 29, 42
 `cumsum`, 127, 132, 271
 `docstring format`, 659–660
 `dtype`, 129, 131, 296
 `exp`, 28
 functions, summary table, 380–382
 `genfromtxt`, 296, 312, 366
 `iinfo`, 276
 importing, *see* importing
 `info`, 32
 `isclose`, 241, 408, 489, 600
 `kurtosis`, 528
 `log10`, 44
 `logical_and`, 407, 416
 `logical_not`, 407, 410, 489
 `logical_or`, 407
 `ma`, *see* `ma`
 `max`, 133, 271

- mean, 413, 505
- median, 505
- meshgrid, 452, 468
- min, 133, 191
- missing values, *see* data analysis,
 - missing values
- ones, 129, 185
- randint, 220, 257
- ravel, 375
- reduce, 416
- reshape, 186, 374
- seed, 218
- shape, 183, 193, 270, 368
- sin, 42, 133, 190, 372
- size, 101, 131, 138, 184,
 - 368, 422
- skew, 528
- std, 505
- sum, 101, 271, 372–374, 413
- tan, 42
- ufunc, 416
- uniform, 220, 242, 257
- universal functions, 416
- vectorize, 678
- where, 410–414, 488, 489, 499, 599
- zeros, 128, 185
- numpydoc, *see* NumPy, docstring
 - format
- object, 546
- objects, 131, 268–290, 536–546,
 - 554–556, 560
 - assignment, *see* assignment
 - attributes, 269, 547, 549–550, 557
 - attributes, naming, 274
 - automated handling, 557–560
 - class, 269
 - class, defining, 546–550
 - constructors, 548
 - contents, 272–274
 - copying, *see* copying
 - creating, 548
 - deleting, 30
 - inheritance, 536–546, 550–553, 562
 - looping through, 338, 417–419
 - methods, 269, 418, 547, 549–550, 557
 - methods, naming, 274
 - overloading, 461–465
 - overriding, 552
 - privacy, 274, 549–550
 - saving to a file, *see* pickle files
- scope, *see* variables, scope
- self, 547
- type, 269, 548, 551
- open, 291, 578
- openpyxl, 577
- operating systems, 331, 337, 342, 347,
 - 579, 637
- operators, 10–13, 105–106, *see also*
 - expressions; variables
- +=, 165, 379, 678
- =, 165, 678
- =, *see* assignment
- ==, 140, 240
- and, 224, 231, 243, 406
- arithmetic, 11, *see also* operators,
 - division
- assignment, *see* assignment
- binary, 11
- comparison, 143, 241, *see also*
 - NumPy, `allclose`; NumPy, `isclose`; operators, `==`
- concatenate, 64, 639
- decrementing, *see* operators, `--`
- division, 11, 12, 427, *see also*
 - operators, arithmetic
- in, 518, 600, 631
- incrementing, *see* operators, `+=`
- is, 316
- modulus, 217, 234
- not, 226, 244, 406
- or, 225, 406
- precedence, 25, 229–230
- unary, 11
- os, 332–337, *see also* directories; files;
 - operating systems; programs;
 - shutil
 - `chdir`, 335, 347
 - `curdir`, 340
 - `getcwd`, 335, 340
 - `listdir`, 340, 347, 508
 - `makedirs`, 336
 - `mkdir`, 335
 - `pardir`, 335
 - path, 332–334
 - `abspath`, 333
 - `basename`, 334
 - `exists`, 336, 341
 - importing, 334
 - `isdir`, 341
 - `isfile`, 341
 - `islink`, 341
- join, 332
- splitext, 334
- remove, 339
- rmdir, 336
- system, 331, 342
- topdown, 647
- walk, 634–637, 641, 647, 678
- packages, *see* modules
- packaging, 698–699
 - build management, 699
 - conda, 699
 - continuous integration, 699–700
 - Cookiecutter, 699
 - pip, 698
 - PyPA, 698
 - PyPI, 698
 - Python Packaging User Guide, 699
- pandas, 490, 571, 611–621
 - |, 617
 - &, 617
 - array syntax, 626, *see also* arrays,
 - array syntax
 - boolean selection, 616–617
 - `concat`, 616
 - data objects, 611
 - `DataFrame`, 490, 572, 576–577,
 - 611–612, 617
 - adding columns, 615
 - adding rows, 616
 - `columns`, 576, 611
 - `groupby`, 620
 - `iloc`, 615
 - `index`, 576, 611
 - `loc`, 613, 616
 - positional index, 612
 - rows, number of, 612
 - selection, rows and/or columns,
 - 612–616
 - `sort_index`, 618
 - `sort_values`, 619
 - `to_excel`, 577
 - `values`, 577, 616
 - `DataFrameGroupBy`, 620
 - `get_group`, 621
 - `groups`, 621
 - grouping, 620–621
 - missing values, *see* data analysis,
 - missing values
 - NaN, *see* NaN
 - `read_excel`, 577

- Series**, 490–492, 573, 576–577, 617
 - fillna**, 492
 - index**, 577
 - mean**, 487
 - name**, 577
 - sort_index**, 618
 - sort_values**, 619
 - str**, 627
 - to_excel**, 577
 - values**, 492
- parameters, 33, 459
 - *args/**kwargs**, 461–465
 - arguments, contrast with, 57
 - dummy variables, 57
 - input, 29
 - checking, 164, 299
 - keyword, 91–93, 462
 - optional, 92
 - positional, 55–58, 74, 75, 86, 462
 - required, 58
 - list, 33
 - module variables, contrast with, 57
 - number, indeterminate, 461–465
 - passing, 290
- pass**, 145
- pattern matching, 341, 600, *see also*
 - glob**, **glob**
- PEP 8, 363, *see also* programming, readability; style
- performance, 414–415, 422–423, 682, 704, 705
 - algorithms, 603, 609
 - asymptotic algorithmic complexity, 666–668, 682
 - bottlenecks, fixing, 678–681
 - in-place operations, 670
 - inefficient code, 668–670
 - just-in-time compilation, *see* just-in-time compilation
 - looping, *see* looping, performance
 - wall clock time, 667
- pickle files, 573, 578–579
- Pillow, 394
- pip**, *see* packaging, **pip**
- plotting, *see* Matplotlib; **cartopy**
- print**, 15, 22, 30, 37, 64, 74, 185, 361, 595
- privacy, *see also* objects, privacy
 - documentation, impact on, 658
- programming, 3–5, 35–41, 138, 155–158, 355–359, 505
 - compound statements, 134
 - flexible, 505–511
 - indentation, *see* indentation
 - more advanced structures, using, 505–511, 521–522
 - object-oriented, *see* objects
 - patterns, 357, 505–511
 - procedural, 268, 554
 - readability, 15, 202, 231, 235, 522, 641, 682, *see also* style
 - testing, *see* testing
- programs, 35–41, 50, 328–331, 505, *see also* C/C++; cython; Fortran; numba; os; performance
 - automated running, 342
 - interactive running, 9, *see also* interpreter
 - non-Python, running, 342–343
- pylint**, *see* testing, linting
- PyPA**, *see* packaging, **PyPA**
- PyPI**, *see* packaging, **PyPI**
- pyplot**, *see* Matplotlib, **pyplot**
- pytest**, *see* testing, **pytest**
- queue**
 - LifoQueue**, 513, *see also* stacks
 - Queue**, 514, *see also* queues
- queues, 513–515, 524
- raise**, 299
- random number
 - seed, 219
- random numbers, 217–221, *see also*
 - NumPy**
 - distributions, 219–221
 - NumPy random**, importing, 221
 - repeat period, 218
 - seed, 218
- range**, 130, 138, 193
- recursion, 340, 633–634, 637–644
 - base case, 638
 - stopping case, 638
- reStructuredText**, 659, 661
- return**, 34, *see also* functions, return value
- scatter plots, *see* Matplotlib, scatter plots
- Scientific Python Cookiecutter, 701
- scikit-image, 394
- scope, *see* variables, scope
- scripting, 35–41, *see also* data analysis, automating; directories; files; programs
- searching, 595–598, *see also* **glob**, **glob**; **pandas**; pattern matching
 - arrays, 599, 600
 - binary, 602–604, 642–643
 - linear, 597, 601–603
 - lists, 598, 599
 - membership testing, 518, 600, 631, *see also* operators, **in**
 - testing in a loop, 598
- selection, *see* slicing
- set**, *see* sets
- setattr**, 557
- sets, 515–516, 525
- shape, *see* arrays, *n*-dimensional, shape; arrays, two-dimensional, shape; **NumPy**, **shape**
- shells, *see* operating systems
- shutil**, 337–340, *see also* operating systems; os
 - copy2**, 338–339
 - copytree**, 339
 - move**, 337–338
 - rmtree**, 340
- slicing, 60–61, 99–101, 615, 648, *see also*
 - pandas**
 - arrays, 99–101, 114, 115, 151–153, 307, 376
 - n*-dimensional, 370–372, 377
 - two-dimensional, 179, 187–189
 - lists, 60–61, 307
 - strings, 64, 507
- sorted**, 518, 553, 601, 609, *see also* sorting
- sorting, 595–598, 618–620, *see also*
 - pandas**; **sorted**
 - insertion, 606–609
 - lists, 601
 - selection, 598, 604–606, 609
- Sphinx**, 660–665
 - autodoc**, 661, 664, 665
 - automodule**, 664
 - directives, 663, 665
 - HTML**, generating, 661
 - quickstart, 661
 - reference guides, autogeneration, 664
 - table of contents, 663

- `toctree`, 663
- Spyder, 38–41
 - current working directory, 335
 - debugging, 362
 - line magic, 55
- stacks, 511–513, 523
- `str`, 65, 104
- strings, 62–66, 343, *see also* pandas, `Series`, `str`
 - attributes, 280, 338
 - concatenate, 64
 - converting to, 65
 - creating, 62–64
 - empty, 64
 - `find`, 323
 - `format`, *see* strings, `format` codes
 - format codes, 282–285, 305, 471
 - formatting, *see* strings, `format` codes
 - `isupper`, 323
 - `join`, 281–282, 294, 646
 - length, 65
 - looping, *see* looping
 - `lower`, 280
 - membership testing, *see* searching, membership testing
 - methods, 280–285, 293–295, 338
 - newline, 64, 264
 - slicing, 64
 - `split`, 281, 293
 - `strip`, 280
 - substring, 65
 - tab, 64, 295
 - `title`, 280
 - `upper`, 280
- style, 15, 208, 223, 363–364, 522, 669, 682, *see also* PEP 8; programming, readability
- subarrays, *see* arrays, sheets; slicing
- sublists, *see* slicing
- `super`, 543, 553
- `sys`
 - `float_info`, 276
 - `getrecursionlimit`, 641
 - `maxint`, 277
- tab character, *see* strings, tab
- terminal, 16, 36–37
- testing, 155–158, 360–363, 690–692, *see also* debugging
 - acceptance testing, 690
 - breakpoints, 361–363
 - cProfile, 672–674
 - integration testing, 690
 - line-profiler, 674–676
 - linting, 683–686
 - memory-profiler, 676–678
 - profiling, 670–678
 - pytest, 688–690
 - system testing, 690
 - Test-Driven Development, 690–692
 - timeit, 671–672
 - unit testing, 686–690
 - unittest, 687–688
- text editors, 36
- text files, 36, 578, 579, *see also* file input/output
- time, 414
- timeit, *see* testing, timeit
- timestep, *see* modeling, timestep
- tox, 700
- truth table, 227, 244–245, *see also* types, booleans
- `try/except`, 298–300, 315–316
- `tuple`, 62
- tuples, 61, 184, 368, *see also* lists
- `type`, 103, 552
- typecodes, *see* arrays, data types
- types, 101, 103–106, 560, *see also* variables
 - arrays, *see* arrays
 - assignment, *see* assignment
 - booleans, 221–230, 428, 488–490
 - `DataFrame`, *see* pandas
 - dictionaries, *see* dictionaries
 - dynamic, 104, 556
 - floating point, 12, 103, 222, 239–241, 274–278
 - integers, 103, 222, 239, 240, 274–278
 - lists, *see* lists
 - objects, *see* objects
 - queues, *see* queue; queues
 - `Series`, *see* pandas
 - sets, *see* sets
 - stacks, *see* queue; stacks
 - strings, *see* strings
 - tuples, *see* tuples
- unittest, *see* testing, unittest
- variables, 9, 13–15, 560, *see also* types
 - accumulator, 14, 149
 - assignment, *see* assignment
 - booleans, 222–224
 - copying, *see* copying
 - deaccumulator, 14
 - deleting, 30
 - global, 559
 - local, 559
 - memory, *see* memory
 - modules, 32
 - naming, *see* PEP 8; programming, readability; style
 - scope, 34, 559
- vectorization, *see* arrays, array syntax
- version control, 693–698
 - branching, 697–698
 - cloning, 696
 - committing, 694, 696, 697
 - forking, 696
 - Git, 694–698, 700
 - merging, 698
 - pull requests, 696, 698
 - pulling changes, 696
 - pushing changes, 696, 698
 - remote repositories, 696–697
 - repository, 694
 - staging, 694, 695
 - status, 695
 - tracking a file, 694
 - workflow, 697
- `while`, *see* looping
- wildcard, 341
- `with/as`, 295
- `yield`, 679