

Index

A

add() method, 60
add_column() method, 412
addTypeEqualityFunc() method, 362, 368
Affero General Public License (AGPL), 375
American Standard Code for Information
Interchange (ASCII), 312–313
annotation_processor(), 134
append() method, 60
Arithmetic operations, 228, 229
ASCII, *see* American Standard Code for
Information Interchange (ASCII)
attach_to_class() method, 403

B

Backward compatibility, 25–26
Berkeley Software Distribution (BSD)
license, 376
Bitwise operations, 234–236
__bool__() method, 227
Bound methods, 205–206
break statement, 43
Built-in function type, 83
Built-in id() function, 270
Built-in type() function, 149
Byte string
chr() and ord(), 306
standard strings, 305
struct module, 307

C

Cache, 29
Callables, 259
__call__() method, 419
chain() function, 56
Classes
attributes
descriptors, 201
functions, 198
“get” and “set” functions, 198,
202, 204
properties, 198
creation
code block, 186–187
individual plugin, 195
metaclasses, 190, 196
namespace controlling, 196
plugin framework, 192
plugin mount class and plugin
subclass, 194
runtime, 188
inheritance
contact class, 163–164
“get” and “set” functions, 162
introspection, 184
MRO (*see* Method resolution order
(MRO))
multiple inheritance, 165
Person and Company classes, 164
Python class, 162

INDEX

Classes (*cont.*)

- iterators, 221–223
 - magic methods
 - attributes, 215
 - automatic subclasses, 213
 - delattr() method, 217
 - setattr() function, 215–216
 - getattr() function, 215
 - instantiation creation, 211
 - repr() method, 220
 - string representations, 218
 - str() function, 218–219
 - methods
 - assigning functions, 210
 - bound methods, 205–206
 - class methods, 207
 - static methods, 209
 - unbound methods, 204–205
- Class methods, 207–209
- clear() method, 62
- coerce_arguments() decorator, 135, 136
- Collections module
- default dictionaries, 67
 - named tuples, 65
 - ordered dictionaries, 66
 - sets
 - add() method, 60
 - append() method, 60
 - clear() method, 62
 - difference() method, 63
 - discard() method, 61
 - in keyword, 59
 - intersection() method, 63
 - issubset() method, 64
 - issuperset() method, 64
 - letters determination, 58
 - pop() method, 61
 - remove() method, 61

- union() method, 62
 - update() method, 60, 62
- Comma Separated Values (CSV) files
- argument checking
 - column name array, 435
 - keyword arguments, 434, 436
 - positional arguments, 433
 - declarative programming (*see* Declarative programming)
 - enumerate() method, 436
 - kwargs, 436
 - looping, 437
 - reader() method
 - Content class, 443–444
 - csv_dialect attribute, 440
 - Dialect class, 441
 - iterable object, 439
 - __iter__() method, 441–442
 - __next__() method, 441–443
 - row class, 439, 440
 - reader objects, 431
 - row class, 438
 - writer() method
 - contents, 448
 - row subclass, 445
 - title() method, 446
 - writerow() method, 445–447
 - writer objects, 432
- Comparison operations, 243
- Context managers, 260
- Control flow
- catching exceptions
 - EnvironmentError, 34
 - except keyword, 32–33
 - IOError, 33
 - OSError, 34
 - raise keyword, 32
 - SystemExit, 33

- try keyword, 32
- TypeError, 34
- conditional expressions
 - and operator, 48
 - false value, 48
 - if/else function, 46
 - or operator, 48
- count_lines() function, 39
- definition, 32
- except blocks, 40
- exception chains, 37
- optimizing loops, 43
- UnicodeDecodeError, 42
- with statement, 44
- copy() method, 297
- CounterMeta class, 419
- count_lines(), 34
- CSV framework, 395
- custom_operator() function, 109
- Cyclical counting, 283–286

D

- Declarative framework, 392–393
- Declarative programming
 - advantage, 394
 - base class, 396
 - class declaration, 394
 - end-user code, 397
 - example.py, 410
 - fields
 - arguments, 401
 - attach_to_class() method, 403
 - base class, 401
 - columns, 401
 - definition, 396
 - __init__() method, 401
 - title, 401

- __init__.py module, 409
- instance methods, 395
- managing options
 - component, 397
 - custom class, 398
 - dialects, 400
 - dictionary implementation, 397
 - encode() method, 399
 - header row, 399
 - __init__() method, 398
 - open() function, 399
 - validation method, 398
- metaclasses (*see* Metaclasses)
- options container, 396
- placeholder, 397
- string variable, 393
- valuable approach, 395
- deepcopy() method, 299–301
- __del__() method, 284–286
- discard() method, 61
- Distribution
 - licensing
 - AGPL, 375
 - BSD license, 376
 - free software
 - foundation, 378
 - GPL, 373
 - LGPL, 376
 - OSI, 378
 - online documentation, 386
 - packaging
 - definition, 378
 - documentation, 378–379
 - MANIFEST.in, 382–383
 - sdist command, 384
 - setup.py, 379
 - tests directory, 379
 - PyPI, 386

INDEX

- divmod() method, 231, 239
- Django and Elixir approach, 392
- Doctest module
 - code formatting, 350
 - docstrings, 350
 - documentation, 353
 - print() function, 352
 - repr() function, 352
 - testmod() function, 354
 - times2() function, 354
- Documentation
 - comments, 333
 - docstrings, 334
 - external
 - installation and configuration, 337
 - reference documents, 337
 - tutorials, 337
 - proper naming, 332
 - utilities
 - formatting, 339
 - links, 340
 - Sphinx, 343
- Don't Repeat Yourself (DRY), 20–21, 31
- Double-asterisk syntax, 400
- dump()/dumps() functions, 290

E

- elif keyword, 7
- __enter__() method, 260
- enumerate() method, 436
- example() function, 289
- __exit__() method, 261

F

- Feedparser, 327–328
- Fibonacci sequence, 28

Field library

- Column class, 422, 423
- DateColumn class
 - date() method, 428
 - format object, 427
 - strptime() method, 428
 - to_python() method, 429–430
- DecimalColumn, 425
- FloatColumn, 424
- IntegerColumn, 424
- StringField, 423
 - to_python() method, 422
 - to_string() method, 422
- __floordiv__(), 230
- Floor division, 230
- Functions
 - annotations
 - annotation_processor, 132
 - coercion, 134
 - decorators, 137
 - typesafe() decorator, 134
 - type safety, 121, 137, 139–143
 - arguments
 - flexibility, 84
 - get_arguments(), 100, 102
 - introspection, 95
 - invoking functions, 92
 - positional arguments, 90
 - preloading, 93
 - Python's keyword, 84
 - required arguments, 89
 - update() function, 101
 - values, 96
 - variable keyword, 87, 92
 - variable positional, 85, 90
 - decorators
 - arguments, 111
 - closures, 107

- creation, 118
- log_error() function, 106
- memoization, 116
- @ syntax, 106
- with/without
 - arguments, 114
- wrappers, 109
- generators, 143
- introspection
 - definition, 148
 - docstrings, 151
 - modules and packages, 150
 - object types, 149
- Lambdas, 146
- programming language, 83
- functools.partial(), 109
- functools.wraps()
 - decorator, 110, 111

G, H

- Garbage collection
 - cyclical references
 - del b, 283
 - __del__() method, 284–286
 - gc.collect(), 285
 - gc.module, 285–286
 - example() function, 289
 - namespace, 289
 - reference counting, 281–283
 - weak reference, 287
- General Public License (GPL), 373
- get() method, 16
- get_arguments(), 99, 100, 103–104
- __getattr__() method, 275
- getdoc() function, 152
- __getitem__() method, 253
- __getstate__() method, 291–292, 295–296

I, J, K

- Image processing
 - changing orientation, 370
 - crop function, 370
 - filters, 371
- __import__() function, 74
- Importing code
 - asterisk, 71
 - fallback, 68
 - __future__ module, 70
 - __import__() function, 74
 - importlib module, 78
 - relative imports, 74
- importlib module, 78
- import_module() function, 78
- import_path() function, 77
- __index__() method, 240
- IndexError, 253
- __init__() method, 271, 273, 362, 368, 401, 405
- insort() method, 417
- inspect.getdoc(), 152
- intersection() method, 63
- Introspection techniques, 31
- __invert__() method, 236
- isinstance() function, 122, 149, 184
- issubset() method, 64
- issuperset() method, 64
- __iter__() method, 244, 249, 442
- Iterables
 - break statement, 245
 - built-in range() function, 245
 - generators, 248
 - __getitem__() method, 247
 - __init__() method, 244
 - __iter__() method, 244
 - __next__() method, 245

INDEX

Iteration

- category, 27
- chaining iterables, 56
- definition, 49
- dictionary comprehensions, 56
- for loop, 50
- generator expressions, 53
- list comprehensions, 52
- looping, 28
- memory allocation, 28
- object-oriented perspective, 28
- range() function, 49
- sequence unpacking, 50
- set comprehensions, 55
- zipping iterables, 57

L

- Lambda function, 416
- Lesser General Public License (LGPL), 376
- List comprehensions, 52–53
- log_error(), 106, 107

M

- map() function, 57
- Mappings, 257
- math.ceil(), 241
- Matplotlib library, 157
- Memoization, 30
- memoize() decorator, 117
- Metaclasses
 - attach_to_class() method, 407
 - class-level attributes, 405
 - encoding, 405
 - hasattr(), 408
 - dialect class, 406
 - inner class, 406
 - loop, 408

RowMeta, 409

Method resolution order (MRO)

- base class, 166
 - breadth-first approach, 171
 - C3 algorithm, 176
 - C3 function, 172, 176
 - candidate class, 171
 - candidate removing, 174
 - candidate selection, 175
 - horizontal approach, 167
 - non-first classes, 173
 - parent classes, 171
 - purchase() method, 167
 - simple two-element list, 170
 - super() function, 179–182
 - test() method, 169, 182–183
 - TypeError, 177
 - while True loop, 173
- min() processes, 53
- min() function, 54
- __mod__() method, 316
- multiply_by(), 108

N

- namedtuple() function, 65
- Namespace dictionary
 - Borg pattern
 - __init__() method, 271, 273
 - __new__() method, 273–275
 - __dict__ attribute, 270
 - self-caching properties
 - cachedproperty(), 278–280
 - __getattr__() method, 275
 - ORM, 276
 - Person.get_employer()
 - method, 277
 - @property, 277–278
 - __setattr__() method, 275

`__new__()` method, 273–275, 420

New BSD license, 377

`__next__()` method, 245, 442

Numbers

comparison operations, 243

`__floor__()` method, 241

`__index__()` method, 240

`__int__()` method, 240

`__round__()` method, 241

sign operations, 242

`TypeError`, 240

NumPy

arrays, 344–346

installation, 344

scientific computing and data

manipulation, 344

statistical functions, 347

O

Object management

Beautiful Soup, 301–303

copying

`built-in sorted()` method, 299

`copy()` method, 297

`deepcopy()` method, 299–301

shallow copy, 297

drawback, 296

garbage collection (*see* Garbage collection)

identity, 269

namespace dictionary (*see* Namespace dictionary)

object type, 269

pickling

`__currency__()` method, 294

`dumps()` function, 290

`__setstate__()` method, 291–292, 295–296

`load()` and `loads()`

functions, 290

`__setstate__()` method, 293

value, 269

Object-Relational Mapping (ORM), 276

`open()` function, 399

Open Source Initiative (OSI), 375

Operations

bitwise, 234

`__bool__()` method, 226

mathematical operations

arithmetic, 228

floor division, 230

modulo operation, 230

`__pow__()` method, 233

rectangle, 226

variations, 236

`OrderedDict` class, 66

Ordered fields

`Column.__init__()`

`attach_to_class()`, 417

`Column` object, 414, 415

`Dialect` object, 416

`__init__()` method, 414, 417

`insert()` method, 417

`RowMeta` class, 416

`Column.__new__()`, 418

`CounterMeta.__call__()`, 419

`DeclarativeMeta.__prepare__()`

`add_column()` method, 412

`Column` class, 414

namespace dictionary, 412

`RowMeta`, 411

options, 421

P, Q

Pareto Principle, 22

Pickle command, 289

INDEX

- pop() method, 61
- __pow__() method, 232–233
- Preloading arguments, 93
- __prepare__() method, 411
- Principles and philosophy, Python
 - backward compatibility, 25
 - DRY, 20
 - loose coupling, 21
 - Pareto Principle, 22
 - Robustness Principle, 23
 - Samurai Principle, 22
 - Zen of Python
 - bracket syntax, 16
 - code blocks, 7
 - code snippet, 6
 - comp.lang.python, 2
 - complexity, 6
 - documentation, 17
 - elif keyword, 7
 - encoding, 15
 - error handling, 14
 - explicit code, 4
 - flat structures, 8
 - get() method, 16
 - implementation, 11, 18
 - interface, 5
 - is_valid code, 3
 - iterative development, 18
 - keyword arguments, 3
 - namespace handling, 19
 - PEP-8, 8
 - readability Counts, 9–10
 - robust error-handling system, 11
 - Unicode strings, 16
 - validate() method, 13
- Pseudo-random number generator, 79
- Python 3.0, 36
- Python Enhancement Proposal (PEP), 2

- Python extensions
 - Beacon components, 79
 - charts type, 158
 - get value, 80
 - Matplotlib library, 157
 - mean() and groupby(), 156
 - NIST Beacon, installation, 80
 - Pandas and Matplotlib,
 - installation, 154
 - Pandas to display data, 155
 - statistics, 154
 - text file, 155
- Python Imaging Library (PIL), 369
- Python Package Index (PyPI), 386
- Python's introspection, 95–96
- Python's keyword
 - arguments, 84

R

- readlines (), 42
- Reference counting, 281–283
- remove() method, 61
- return statement, 43
- returns() function, 143
- Robustness principle, 23
- __round__() method, 241

S

- Samurai principle, 22
- Scrapy
 - installation, 263
 - project setup, 263, 265
 - retrieve web data, 265
 - running, 263
 - shell options, 265–266
 - web page, view, 265

- Secrets module
 - password generation, 388–389
 - random number algorithm, 387, 388
 - self.fail(), 368
 - Sequences
 - additional attributes and behaviors, 250
 - built-in len() function, 251
 - class range, 256
 - __contains__() method, 256
 - __delitem__() method, 255
 - __getitem__() method, 252–253
 - __len__() method, 251
 - reversed() function, 251
 - __setitem__() method, 255
 - Sequence structure, 32
 - __setattr__() method, 275
 - __setitem__() method, 255
 - __setstate__() method, 293
 - setup() function, 379
 - Shallow copies, 297–299
 - Sign operations, 242
 - Simplified BSD license, 377
 - sort() method call, 146
 - sort() operation, 416
 - Sphinx, 343
 - startswith() method, 13
 - Static methods, 209
 - str() function, 316
 - strftime() method, 430
 - String representation, 218
 - Strings
 - bytes
 - chr() and ord(), 306
 - standard strings, 305
 - struct module, 307
 - formatting
 - custom format, 326
 - customizable validation function, 319
 - explicit format() method, 319
 - format(), 320
 - keyword arguments, 319
 - object reference, 321
 - plain text table of contents, 324
 - positional arguments, 319
 - __repr__() method, 321
 - standard format, 322
 - __str__() method, 321
 - simple substitution, 316
 - text
 - encodings, 313
 - unicode, 312
 - strftime() method, 428
 - Struct module, 307
 - subclass() function, 185
 - suppress_errors() decorator, 109, 112, 119
 - suppress_errors() function, 106, 113
 - symmetric_difference() method, 64
- ## T
- tearDown() method, 367
 - Testing
 - custom test class, 368
 - definition, 349
 - doctest module
 - code formatting, 350
 - docstrings, 350
 - documentation, 353
 - print() function, 352
 - repr() function, 352
 - testmod() function, 354
 - times2() function, 354
 - test-driven development, 349–350
 - unittest module
 - addTypeEqualityFunc()
 - method, 362

INDEX

Testing (*cont.*)

- assertEqual(), 361
- assertion, 358
- AssertionError, 360
- assertRaises(), 364–365
- assertRaisesRegexp(), 365
- comparison methods, 363
- identity, 366
- main() function, 358
- sequence content, 364
- setUp() method, 356
- strings, 363
- tearDown() method, 367
- TestCase class, 356
- tests.py, 359
- with block, 365

testmod() function, 354

testNumber() method, 358

tests.py module, 359

test_value() function, 48

title() method, 446

to_python() method, 422

to_string() method, 422

Transmission Control
 Protocol (TCP), 23

Transparency, 30–31

__truediv__() method, 229

typesafe() decorator, 140–141

typesafe() function, 126, 129, 143

U

Unbound methods, 204–205

union() method, 62

Unittest module

- addTypeEqualityFunc() method, 362
- assertEqual(), 361
- assertion, 358

- AssertionError, 360
- assertRaises(), 364–365
- assertRaisesRegexp(), 365
- comparison methods, 363
- identity, 366
- main() function, 358
- sequence content, 364
- setUp() method, 356
- strings, 363
- tearDown() method, 367
- TestCase class, 356
- tests.py, 359
- with block, 365

update() function, 101

update() method, 60, 62

UTF-8, 314

V

validate() method, 13

Variable arguments, 89, 92, 104

Variable keyword arguments, 92

W

Weak references, 287–289

What You See Is What You Mean
 (WYSIWYM) approach, 338

Wrapped function, 109

writerows() method, 446–447

X, Y

XOR operation, 64

Z

zip() function, 57