

INDEX

Symbols & Numbers

`__import__`, 16–17
`__init__.py`, 8
`__slots__`, 163–167
`__repr__`, 211

A

abstract syntax tree (AST),
 135–141, 147
 walking through, 139
Advanced Message Queuing Protocol
 (AMQP), 184–186
aiohttp library, 183
all() function, 128
ambiguous times, 55
any() function, 128
API (application programming
 interface)
 designing, 45
 documentation, 41–42, 46–47
 managing changes, 40–41, 46
architecture
 event-driven, 181
 service-oriented, 184
AST (abstract syntax tree),
 135–141, 147
asyncio module, 182–184
attr module, 210–212

B

bisect module, 159–160
 bisect.bisect() function, 159
 bisect.bisect_left() function, 160
 bisect.insort() function, 160
buffer protocol, 170–174
bytearray, 173–174

C

C10K, 181
cache, 167–168
CLOS (Common Lisp Object System),
 203–205
closure, 159
Coghlan, Nick, 74
collections module, 153
 Counter() method, 154
 defaultdict, 153
 namedtuple class, 165
Collins, Robert, 97–98
Common Lisp Object System (CLOS),
 203–205
console scripts, 69–70
contextlib, 208–210
context management protocol, 207–210
context managers, 207–210
copy.deepcopy(), 176
Counter(), 154
coverage tool, 88–89
cProfile module, 154–155
CPython, 163, 169, 176, 178–179

D

databases, 187–199
 backends, 190
 existing time zones, 52
 relational database management
 system (RDBMS), 187,
 195–197
data structures, 152–154
datetime, 50–51, 54–55
dateutil, 52–53, 55
debtcollector library, 14, 44
decorators, 100–107, 142–143
 class decorators, 103
 creation, 100–101
 stacking, 102, 103

- defaultdict, 153
- de Vienne, Christophe, 45–47
- dis module, 156–158
 - dis.dis() function, 156
- distribution, 57–74
 - building *setup.py*, 58–59
 - format, 61
 - packaging with *setup.cfg*, 60–61
 - Wheel standard, 61–63
- distutils library, 58
- doctest module, 38
- documentation, 34–35

E

- entry points, 67
 - visualization, 68
- enumerate() function, 127
- event-driven architecture, 181–182

F

- filter() function, 127
- first() function, 130–131
- fixtures, 81–82
- flake8, 12, 95, 140–141
- Fontaine, Dimitri, 195–199
- frameworks, 26–27, 31
- functional programming, 119–121
- functools module, 105
 - partial() method, 131–132
 - update_wrapper() function, 105
 - wraps, 106

G

- generators, 121–123
 - inspecting, 124–125
- generic methods, 205
- GitHub, 35
- global interpreter lock (GIL), 13, 169, 176, 178

H

- Harlow, Joshua, 13–14
- Hellmann, Doug, 27–31
- hierarchy, 7
- Hy, 18, 145–149

I

- import hook, 18
- import keyword, 16–17
- inspect module, 106–107, 124
- interprocess communication, 185–186
- iso8601 module, 54–55
- itertools module, 132–133

J

- JSON, 191, 193–194
- just-in-time (JIT) compilation, 14, 169
- Jython, 178

K

- KCacheGrind, 155–156

L

- lambda() function, 131
- layout, 7
- least recently used (LRU) cache, 167–168
- libraries, 15
 - API, 46
 - external, 22, 23, 26
 - standard, 20, 28–29
- Lisp, 145–146, 147–148, 203
- list comprehension (listcomp), 125–126

M

- map() method, 127
- memoization, 167–168
- memoryview, 171
- meta path finder, 19–20
- method resolution order (MRO), 115
- methods, 107–117
 - abstract, 110–113
 - class, 109–110, 112–113
 - generic, 205–207
 - mixing, 112–113
 - static, 108–109, 112–113
- mock library, 84–88
- modernize module, 203
- modules included in the standard library, 21
- multiple inheritance, 114
- multiprocessing, 179–181, 185
- multithreading, 178–181

N

namedtuple class, 165–166
next() function, 121–122

O

object relational mapping (ORM),
188, 197–198
OpenStack, 1, 13–14, 22, 29, 97
optimization, 151–174
ordered lists, 159

P

packaging solutions, 74
pbr (Python Build Reasonableness), 60
PEP (Python Enhancement Proposal)
 PEP 440, 8–10
 PEP 7, 11
 PEP 8, 10–12
 pep8, 10
pip, 24–26
plugins, 71
poll() function, 181
PostgreSQL, 190–194, 195–196,
198–199
profiling, 154
psycogp2 library, 192
pure functions, 120
pyflakes, 12
pylint, 12
PyPI, 24, 64–67
pyprof2calltree, 155
PyPy, 169
pytest, 76–81
 coverage, 88
 fixtures, 81
 mark, 80
 pattern, 79
 parallel, 81–82
 scenarios, 83
PYTHONPATH, 18
Python versions, 5–6, 30, 201–203
 Python 2, 6
 Python 3, 6, 13, 23, 27, 30

R

relational database management
 system (RDBMS), 187,
195–197

REpresentational State Transfer
 (REST), 184
reStructured Text (reST), 34–36

S

scaling, 177–186
scenarios, 83
select() function, 181–182
semantic versioning, 9
service-oriented architecture, 184
setup.cfg, 59–61
setup.py, 7, 57–61
setuptools library, 58–59, 67
singledispatch() function, 205–207
Single Responsibility Principle
 (SRP), 30
six module, 201–202
sockets, 172–173
sorted() function, 128
sorted list, 159–160
Sphinx, 33–40, 42
 autodoc, 36–37
 doctest, 38–39
SQL, 187–190, 197–198. *See also*
 PostgreSQL
SQLAlchemy, 22, 30, 190
Stinner, Victor, 174–176
streaming, 190
strings, 202
super() method, 114–117
sys module, 17
sys.path variable, 18

T

Tagliamonte, Paul, 147–149
taskflow, 14
testing
 policy, 96, 97–98
 skipping, 78
 unit, 75–76
threads, 178
timeit module, 175
timestamps, 49–56
time zones, 49–50, 52–54
tox, 92–96
tox-travis, 97
Travis CI, 96

U

Unicode, 202
update_wrapper() function, 105

V

versions
 API, 41
 numbering, 8–10
 Python, 5, 95
virtual environments, 90–96
 re-creating, 94
 setting up, 91–92
 tox, 92–93

W

warnings, 43–44
Web Server Gateway Interface
 (WSGI), 29
Wheel, 61–63
 universal, 63
with, 207
wraps decorator, 106

Y

yield, 121–123

Z

zero copy, 170
ZeroMQ, 185–186
zip() function, 129