

# Index

## A

- absolute paths, Buildbot, 124
- accept.py build tool, 358
- acceptance testing, 339, 340
  - build to include acceptance tests, 353
  - converting URLs into feed objects, 188
  - creating directory for, 346
  - description, 140, 173, 339
  - Framework for Integrated Tests (FIT), 340–353, 367
    - creating column fixture test with FIT, 346–353
  - observing failing test first, 152
- PyFit, 340–353
  - writing requirements, 341–344
- RSS reader application, 150, 152
- running FIT with Buildbot, 353–367
  - defining Buildbot configuration file, 353
  - getting regular builds, 366–367
  - making reports available, 358–366
  - preparing slave, 353
  - running builder, 354–357
- running with every build, 339, 346
- using data files, 189
- active record pattern
  - ORMs, 266
  - SQLObject, 267
- add (\_\_add\_\_) method, 193
- add command, Subversion, 50
- add method, pMock, 201
- add method, PyMock, 221
- adding files, Subversion
  - from command line, 50
  - through Eclipse, 68
- addRemoveName keyword
  - multiple relationships, SQLObject, 282
- addTestPage/addTestSuite methods
  - aggregating JsUnit tests, 335–336
- add\_single\_feed method, PyMock, 216–217
  - refactoring, 218
- admin\_pw keyword, DBMigrate, 301
- admin\_user keyword, DBMigrate, 301
- AggregateFeed class, pMock
  - combine\_feeds method, 199
  - creating FeedEntry factory, 200
  - empty AggregateFeed output, 209
  - entries initialized to Set, 201
  - feeds\_from\_urls method, 203
  - initializing FeedParser factory, 203
  - refactoring/extracting, 198
  - reimplementing from\_urls method, 204
- AggregateFeed class, PyMock
  - entries initialized to Set, 222
  - print\_entry\_listings with empty feeds, 225
- aggregating JsUnit tests, 335–336
- aggregating two feeds, 194–195
- aggregator, 149
- agile development, 1, 3, 4
  - databases and, 264, 296, 306
  - impact on DBA organization, 264
  - using IDEs with, 21
- agile host, aliasing, 104
- agile methods, 4
  - collective code ownership, 12–13
  - continuous integration, 16–17
  - continuous reflection, 15
  - documentation, 17
  - on-site customers, 8–9
  - pair programming, 5–7
  - refactoring, 11–12
  - short iterations, 13–15
  - simple design, 12
  - system metaphor, 8
  - test-driven development, 10–11
  - unit tests, 9
  - user stories, 7–8
- Ajax (Asynchronous JavaScript and XML), 312
- aliasing hosts, 104
- all (\_\_all\_\_) attribute, 182

- all method, SQLAlchemy, 288
  - allow-hosts option, easy\_install, 132
  - altinstall.pth file, 85–86
  - Ant Builder, 164
  - Apache plug-ins
    - mod\_python, 314
  - application class, 175
  - application frameworks
    - connections to web servers, 313
  - application object, 315
  - application servers, Python, 313
  - application tests
    - creating, 175
    - implementing as native Nose tests, 176
  - applications
    - full stack applications, 313
    - web application frameworks, 313
    - web applications, 312–320
  - arguments
    - how dependencies arise, 190
    - mocking calls to self, pMock, 196
  - Arguments field
    - builder properties window, 165
    - External Tools dialog, Eclipse, 169
  - argv file, sys
    - running unit tests manually in Eclipse, 154
  - assert expression, Python
    - success or failure of unit tests, 147
  - AssertionError, 147
  - assertStdoutEquals method
    - running unit tests manually in Eclipse, 158
  - assert\_equals method, 176
  - atoms
    - creating migrations, DBMigrate, 302
  - attribute access, Python, 193
  - attribute defaults, SQLAlchemy, 272
  - attribute setter mocking, PyMock, 229
  - attributes, HTML elements, 310
  - attributes, Nose, 160, 174
  - at\_least playback count, PyMock, 229
  - at\_least\_once calling policy, pMock, 229
  - authority, organizational/technical, 245
  - autoflush attribute, SQLAlchemy, 286
  - autoformatting, 251
  - autoRun parameter, JUnit
    - running tests by URL, 337
- ## ■ B
- backref keyword, SQLAlchemy, 294
  - baseURL property, Subversion, 119
  - bdist\_egg command, Setuptools, 90
  - BeautifulSoup package, 320
  - Bicycle Repair Man program, 39
  - bin directory, Python
    - Install step, Buildbot, 125
  - bin\_dir function, PyFit, 352
  - Block Comments panel, Pydev, 252
  - branch coverage, 238
  - branches
    - revision control systems, 44
  - browsers *see* web browsers
  - build automation
    - continual verification of builds, 103
    - replicable builds, 81
  - build command, Setuptools, 89
  - build directory
    - building projects with setup.py, 88
    - paths used on build slave, 124
    - Subversion ignoring files, 98
  - build factories
    - description, 109, 136
    - encapsulating builder factory in function, 129
    - running FIT with Buildbot, 353, 356
    - supporting Python 2.4 builds, 129
    - using source code, 119
  - build master, Buildbot, 107–112
    - accessing Subversion repository, 116–118
    - build slave contacting, 107
    - configuring, 107
    - description, 104, 136
    - naming, 104
    - options for triggering builds, 104
    - running FIT with Buildbot, 354–355, 357
  - Build Options tab
    - builder properties window, 165
  - build servers, Buildbot, 103–116
  - build slave, Buildbot, 112–116
    - accessing Subversion repository, 116–118
    - checking out code, 120
    - configuring, 112
    - contacting build master, 107
    - description, 104, 136

- installing Buildbot on buildmaster, 104
- naming, 104
- paths used on, 124
- Python installation, 122
- running FIT with Buildbot, 353
- build steps, Buildbot
  - colors used for, 114, 116
  - Compile step, 124–125
  - description, 136
  - enhancing step progress description, 134
  - Install step, 125–128
  - running FIT with Buildbot, 353
  - SVN step, 119–121
- build user, Buildbot
  - configuring build master, 107
  - configuring build slave, 112
- Buildbot, 104–116
  - architecture, 104
  - build master, 104, 107–112
  - build slave, 104, 112–116
  - code coverage report, 259
  - Compile step, 124–125
  - configuring build system, 106
  - contrib directory, 121
  - coverage package, 258
  - enhancing step progress description, 134–136
  - hooks to Subversion events, 121
  - Install step, 125–128
  - installing, 104–106
  - installing contributed programs, 106
  - installing on Windows systems, 107
  - landing page, 111
  - notifications, 121
  - options for triggering builds, 104
  - paths used on build slave, 124
  - reconfig command, 119
  - reconfiguration, 120
  - running FIT with Buildbot, 353–367
    - defining Buildbot configuration file, 353
    - getting regular builds, 366–367
    - making reports available, 358–366
    - preparing slave, 353
    - running builder, 354–357
  - running full test suite at build time, 171–173
  - sendchange command, 114, 120
  - summary, 136
  - supporting Python 2.4 builds, 128–132
  - SVN step, 119–121
  - waterfall display, 111
- buildbot binary
  - server startup script, 112
- Buildbot.tac file
  - configuring build master, 108
  - configuring build slave, 113
- buildbotUrl property, Buildbot, 110
- builder
  - running FIT with Buildbot, 353, 354–357
- builder directory
  - packaging reports, 359
  - paths used on build slave, 124
- builder properties window, 164
  - tabs and fields, 165
- builders, 109
  - preventing overlapping, 122
- Builders menu item
  - project properties window, 163
- builders property, Buildbot
  - configuring build master, 109
  - supporting Python 2.4 builds, 130
- buildmaster
  - aliasing hosts, 104
  - configuring build slave, Buildbot, 112
  - configuring build system, 106
  - installing Buildbot on, 104
- BuildmasterConfig dictionary, 108
- builds
  - build to include acceptance tests, 353
  - building projects with setup.py, 88–91
  - continual verification of, 103
  - continuous builds, 340
  - formal builds, 340
  - incremental builds, Eclipse, 163
  - replicable builds, 85–86
    - path manipulation solution, 85
    - virtual Python solution, 85
  - running full test suite at build time, 171–173
  - supporting Python 2.4 builds, 128–132
  - when unit tests need to be run, 163
- BuildSlave objects, Buildbot
  - configuring build master, 108
  - configuring build slave, 112

build\_ext module, Setuptools, 89  
 build\_py module, Setuptools, 89, 90

## C

callbacks  
   using write() callback, 315  
 camel case  
   naming standards, 248  
 cascade keyword, SQLAlchemy, 295  
 Cascading Style Sheets (CSS), 310, 311  
 centralized revision control systems, 42  
 CGI (Common Gateway Interface), 313  
 chained expressions, SQLAlchemy, 290  
 checkout command, Subversion, 47  
 cheese shop, Python, 84  
 classes  
   creating application class, 175  
   creating Python class, 33  
   subclassing, 193  
 code analysis features, Pydev, 253  
 code coverage, 237–239  
   branch coverage, 238  
   low test coverage, 238  
   patching code coverage report into  
     Buildbot, 259  
   statement coverage, 237  
 Code Formatter panel, Pydev, 252  
 code reviews, 5, 249  
 code style options, Pydev, 252  
 coding  
   collective code ownership, 12–13  
   validating committed code, 256–257  
 coding conventions, 244  
   consistency, 245  
   features assisting writing/analyzing code,  
     251  
   feedback on defective code, 251  
   naming standards, 244  
   Python, 246  
 coding standards  
   autoformatting, 251  
   code analysis features, 253  
   rewarding good code, 248–249  
   templates, Pydev, 254  
 cohesion, 141  
   unit tests illustrating, 142  
 collective code ownership, 12–13  
   unit tests, 143  
 column fixtures, 346  
   representing business processes, 347  
   tests created with FIT, 346–353  
 combine\_feeds method, pMock  
   AggregateFeed class, 199  
   mocking calls to self, 196  
   reimplementing from\_urls method, 204  
 commit command, Subversion, 50  
 commit event, Subversion, 121  
 Commit window  
   working with Subversion through Eclipse,  
     70  
 committers  
   adding to Subversion group, 117  
   build master/slave accessing Subversion,  
     116  
   file ownership, 116  
 Common Gateway Interface (CGI), 313  
 Common tab  
   External Tools dialog, Eclipse, 169  
 communication, developers, 250  
 Compile step, Buildbot, 124–125  
 complexity measurements, 239–242  
 components  
   isolating components under test, 190–192  
 configure step  
   supporting Python 2.4 builds, 128  
 connections  
   application frameworks to web servers,  
     313  
 connect\_application method  
   running DBMigrate with unit tests, 305  
 consistency  
   coding conventions, 245  
 Console Encoding field  
   Common tab, External Tools dialog, 170  
 Console view  
   working with Subversion through Eclipse,  
     65  
 console\_scripts key, setup.py, 91  
 continuous builds, 340  
 continuous integration, 16–17  
   replicable builds, 85  
 continuous reflection, 15  
 contrib directory, Buildbot, 121  
 copy command, Subversion, 51  
   working with Subversion through Eclipse,  
     78

- copying files, Subversion
  - working from command line, 51
  - working through Eclipse, 78
- coupling, 141, 142
- coverage package, 258
  - coverage report weakness, 259
  - running coverage through Nose, 258
  - with-coverage option, nosetests, 258
- createSummary method
  - publishing reports, 361
- create\_entry method, pMock, 200
- create\_entry method, PyMock, 217, 218, 220, 221
- create\_schema method
  - schema definition, SQLAlchemy, 272
- crontab command, 112
- CSS (Cascading Style Sheets), 310, 311
- customer tests, 139, 173
- cyclomatic complexity, 239, 241

## ■ D

- d option, unzip command
  - publishing reports, 360
- daemon option, svnserve, 117
- data mapper pattern, ORMs, 266
- database administrators (DBAs), 263, 265
- database migrations
  - testing databases, 298
- databases
  - adjusting between database environments, 296
  - agile development, 264, 296, 306
  - background, 306
  - background to working with, 263
  - connecting to databases
    - concealing data access, 265
    - DBMigrate, 301
    - SQLAlchemy, 283
    - SQLObject, 268
  - creating rows, SQLAlchemy, 269–272
  - evolution of database design, 264
  - isolation, 264
  - migrations, 298–306
    - DBMigrate, 300–306
    - generating migration instructions, 299
    - locating migration mechanism, 300
    - numbering migrations and playback, 299

- object-relational mappers (ORMs),
  - 265–267
  - Python, 267–296
  - SQLAlchemy, 283–296
  - SQLObject, 267–282
- refactoring, 264, 298
- testing, 264
- db keyword, DBMigrate, 301
- DBAs (database administrators)
  - working with databases, 263
  - working with developers, 265
- DBMigrate, 300–306
  - admin\_pw keyword, 301
  - admin\_user keyword, 301
  - atoms, 302
  - connecting to database, 301
  - creating migrations, 301–303
  - db keyword, 301
  - functions, 302
  - host keyword, 301
  - installing, 300
  - manually migrating database, 303–304
  - migration dictionary, 303
  - port keyword, 301
  - pw keyword, 301
  - revision flag, 304
  - running DBMigrate from program, 305
  - running DBMigrate with unit tests, 305
  - scheme keyword, 301
  - socket keyword, 301
  - string expansion, 302
  - user keyword, 301
  - verbose flag, 304
  - versionable keyword, 301
- debugging
  - unit testing to reduce, 141
  - value of debuggers, 141
- decorators
  - creating rows, SQLAlchemy, 271
- decorators, Python, 161
  - use\_pymock decorator, PyMock, 212
- default attribute, SQLAlchemy, 273
- defaultBranch property, Subversion, 119
- delete command, Subversion, 52
  - working with Subversion through Eclipse, 77
- delete method, SQLAlchemy, 295

- deleting files, Subversion
  - working from command line, 52
  - working through Eclipse, 76
- deleting rows, SQLAlchemy, 275
- dependencies
  - application initializing dependencies,
    - pMock, 211
  - ensuring local dependency processing,
    - 132
  - how dependencies arise, 190
  - isolating components for testing, 191
  - Python packages, 83
  - Setuptools managing dependencies,
    - 92–94
  - testing and, 142
  - test\_rsreader\_dependency\_initialization
    - test, PyMock, 227
  - verifying dependencies initialized
    - correctly, pMock, 206
- description/descriptionDone keywords,
  - Buildbot, 134
- design
  - iterative design methods, 175
  - simple design, 12
- destroySelf method
  - deleting rows, SQLAlchemy, 275
- developers
  - communication, 250
  - kinds of feedback for development, 234
  - pair programming, 249
  - rewarding good code, 248–249
  - why developers need feedback, 234
  - working with databases, 263
  - working with DBAs, 265
- development
  - see also* agile development
  - code reviews, 249
  - goal of development organization, 263
  - using production database, 297
- development environments, 22
- development methodologies
  - agile development, 1
  - iterative methodologies, 3
  - reasons for, 1
  - test-driven development (TDD), 10–11,
    - 146–147
  - waterfall methodology, 2
- development mode, Setuptools, 100–102
- development processes
  - benefits of version/revision control, 41
  - collective code ownership, 12–13
  - continuous integration, 16–17
  - continuous reflection, 15
  - documentation, 17
  - on-site customers, 8–9
  - pair programming, 5–7
  - refactoring, 11–12
  - short iterations, 13
  - system metaphor, 8
  - user stories, 7–8
- development velocity, 242–243
- diff command, Subversion, 54–56
- directories, Eclipse, 38
- directory structure, Subversion, 44
- disconnect\_application method
  - running DBMigrate with unit tests, 305
- Display in favorites menu field
  - Common tab, External Tools dialog, 170
- dist directory
  - Setuptools, 91
  - Subversion ignoring files, 98
- distributed mode, JsUnit, 337
- distributed revision control systems, 42
- Distutils library, 81
  - description, 102
  - installing Python packages, 83
- distutils.cfg file, 86
- Docstrings panel, Pydev, 252
- Document Object Model (DOM), 328
- documentation
  - agile methods, 17
  - developer communication, 250
- docutils package
  - Install step, Buildbot, 126
- DOM (Document Object Model), 328
- DSL (domain-specific languages), 193
- duck typing, 193
- dummies, 191
  - writing in Python, 192
- dummy objects, PyMock, 216

**E**

- EasyMock
  - record-replay, 193
- easy\_install program, Setuptools, 87
  - allow-hosts option, 132
  - development mode, 100
  - installing Buildbot, 104, 106
  - installing from local copy, 96
  - removing existing package, 95
- echo step output
  - configuring build slave, Buildbot, 115
- Eclipse
  - advantages of, 21
  - creating Python development environment, 40
  - development environments, 22
  - directories, 38
  - ignoring files, 100
  - importing from Subversion, 60–64
  - incremental builds, 163
  - installing, 25
  - installing Mylyn, 25–31
  - installing plug-ins, 30
  - installing Pydev, 31–32
  - job management system, 21
  - perspectives, 24
  - project file, 38
  - pydevproject file, 38
  - revision control plug-ins, 59
  - revision control systems and, 59–64
  - running full test suite in development, 168
  - running unit tests manually in, 151–159
  - selecting workspace root, 23
  - shared projects, 59
  - sharing subverted project, 59–60
  - source folders, 33
  - src directory, 39
  - startup screen, 24
  - views, 24
  - workbench, 25
  - working with Python, 25
  - working with Subversion through, 64–79
- Eclipse plug-ins
  - Pydev, 22, 25
  - Pydev Extensions, 22
  - SQLExplorer, 22
  - Subversive, 22
- Eclipse Preferences window, 251
- edit-and-merge process
  - revision control systems, 43
  - Subversion and, 79
- editing files
  - working with Subversion through Eclipse, 71
- eggs
  - installing, 84
  - installing Setuptools, 86
  - naming structure, 84
  - setup.py creating, 87
  - Setuptools and, 81
- egg\_info command, Setuptools, 90
- elements, HTML, 310
- ElementTree package
  - MVC testing markup, 318–319
  - XPath operations supported by, 319
- embedded code
  - testing databases, 298
- encapsulation, 141
- encryption
  - https scheme, 311
- ending keyword, PyMock, 230
- entities, global
  - how dependencies arise, 191
- entry\_listings method, pMock, 207–208
- entry\_listings method, PyMock, 223
- entry\_points attribute, setup.py, 91
- environ dictionary
  - WSGI conversations, 315
- eq constraint, PyMock, 213
- errors
  - branch coverage, 238
  - JUnit, 331
- estimates
  - short iterations, 14
- exceptions, pMock/PyMock, 228
- exclusive locking
  - revision control systems, 43
- executables
  - Install step, Buildbot, 125
  - paths used on build slave, 124
- expectations, PyMock
  - alternative syntax to define, 214
- expected\_items value, 182
- expected\_line value, 176



- expects clause
  - defining expectations, PyMock, 215
  - pMock, 196
  - playback counts, 229
- exploratory testing, 140
- extent, unit tests, 145
- External Tools dialog, Eclipse, 168–169
- external\_requirements attribute, setup.py, 92
- ez\_setup.py program
  - bootstrapping Setuptools, 97
  - installing Setuptools, 86
  - using local copy of Setuptools, 132

## F

- factories *see* build factories
- failures, assertions
  - JUnit, 331
- fakes, 192
- Feature License screen
  - installing Mylyn, 27
- Feature Verification screen
  - installing Mylyn, 30
- feedback
  - defective code, 251
  - environmental feedback, 234
  - kinds of feedback for development, 234
  - measurements, 235–236, 261
  - social feedback, 234
  - why developers need feedback, 234
- FeedEntry constructor, PyMock
  - mocking `__init__` directly, 222
  - test\_feed\_entry\_constructor test, 223
  - test\_feed\_entry\_listing test, 222
- FeedEntry factory, pMock
  - AggregateFeed creating, 200
  - from\_parsed\_feed method, 202
  - listing method, 203
  - test\_feed\_entry\_listing test, 202
  - verifying operation, 202
- FeedParser factory, pMock
  - AggregateFeed initializing, 203
- FeedParser package, 149
  - converting URLs into feed objects, 185
- feeds
  - add\_single\_feed method, PyMock, 216, 217
  - aggregating two feeds, 194–195
  - converting URLs into feed objects, 185, 188
  - creating aggregate entries for feeds, pMock, 197
  - formatting feed entry listings, pMock, 207
  - from\_urls method, pMock, 204
  - taking and converting, pMock, 200
- feeds\_from\_urls method, pMock, 204
  - AggregateFeed initializing FeedParser factory, 203
  - FeedEntry factory, 203
  - reimplementing from\_urls method, 204
- FeedWriter class, pMock
  - formatting feed entry listings, 207
  - initializing stdout attribute, 209
- feed\_from\_url method
  - converting URLs into feed objects, 185
  - isolating components, 190
- feed\_listing method, 180
- fields
  - updating fields, SQLAlchemy, 274
- file repository *see* repository
- file revisions, storage of, 40
- file scheme, 311
- File Types panel, Pydev, 252
- files
  - job management system, 21
- FileUpload build step
  - retrieving reports, 360
- filter method, SQLAlchemy, 289
- filter\_by method, SQLAlchemy, 290
- find-links option, Setuptools
  - fixing options with setup.cfg, 97
- findall method
  - MVC testing markup, 319, 320
- findtext method
  - MVC testing markup, 319
- find\_dev\_root function
  - running PyFit, 352
- find\_packages function, Setuptools, 88
- first method, SQLAlchemy, 288
- FIT (Framework for Integrated Tests), 340–367
  - background, 340
  - components, 340
  - creating column fixture test with, 346–353
  - description, 367



- families of tests created with, 346
- FIT report for broken test, 344
- FitNesse, 340, 368
- fixtures, 341
  - PyFit and, 368
  - PyFit example, 344–345
- reports, 341
- requirement documents, 340, 341, 346
  - writing requirements, 341–344
- running FIT with Buildbot, 353–367
  - defining Buildbot configuration file, 353
  - getting regular builds, 366–367
  - making reports available, 358–366
  - preparing slave, 353
  - running builder, 354–357
- showing successful/unsuccessful test results, 344
- test runners, 341
- XML summary document produced, 343
- FitNesse, 340, 368
- fixtures
  - column fixtures, 346
  - creating column fixture test with FIT, 346–353
  - description, 348
  - Framework for Integrated Tests (FIT), 341
    - writing HTML spec document, 343
  - parts of, 348
- flushing
  - autoflush attribute, SQLAlchemy, 286
- FolderRunner.py
  - creating column fixture test with FIT, 349
  - running PyFit, 351
- foreign keys, SQLAlchemy, 291, 292
- foreign keys, SQLAlchemy, 275
- formal builds, 340
- Framework for Integrated Tests *see* FIT
- frameworks
  - web application frameworks, 337
- from\_parsed\_feed method, pMock
  - FeedEntry factory, 202
  - formatting feed entry listings, 208
- from\_statement method, SQLAlchemy, 290
- from\_urls method, pMock, 204
- from\_urls method, PyMock, 215–216, 219
- full stack applications, 313

- functional testing
  - databases, 298
  - description, 139, 339
  - Framework for Integrated Tests (FIT), 340–353, 367
  - running FIT with Buildbot, 353–367
  - running tests with every build, 340
  - size of functional test suite, 353

- functions
  - creating migrations, DBMigrate, 302
  - decorators, 161
  - how dependencies arise, 190

## ■ G

- generator mocking, PyMock, 230
- get method
  - retrieving objects, SQLAlchemy, 273
- getElementById method, JavaScript, 335
- getitem (\_\_getitem\_\_) function, 193
  - PyMock, 213
- getProcessValue method, Twisted, 364
- getProperty method, BuildStep class, 361
- global entities
  - how dependencies arise, 191
- graphics
  - MVC testing web applications, 317
- groups
  - adding committers to Subversion group, 117
- gui\_scripts key, setup.py, 91

## ■ H

- harness *see* test harness
- hooks
  - hooks sending notifications, Buildbot, 121
  - main method, 175
  - precommit hooks, 256
- host keyword, DBMigrate, 301
- hosts
  - aliasing hosts, 104
  - allow-hosts option, easy\_install, 132
- HTML (Hypertext Markup Language), 310–311
  - HTML documents and forms, 337
  - SGML and, 310
  - Web and, 309, 337
  - writing HTML spec document, 342

- html.WebStatus class
  - configuring build master, Buildbot, 110
- HTMLParser library
  - MVC testing markup, 317
- HTTP (Hypertext Transfer Protocol), 312
  - state, 312
  - Web and, 309, 337
- http scheme, 311
- https scheme, 311
- http\_port keyword
  - configuring build master, Buildbot, 110

**I**

- IDEs
  - development environments, 22
- ignore property, Subversion, 99
- ignoring files, Eclipse, 100
- ignoring files, Subversion, 98–100
- images
  - MVC testing web applications, 317
- impersonators *see* impostors
- import command, Subversion, 46
- Import project window
  - importing from Subversion, 60
- imports
  - monkeypatching and, 186
- impostors
  - categories of, 191
  - description, 231
  - dummies, 191
  - fakes, 192
  - mock objects, 192
  - mocking to break dependencies, 191
  - stubs, 192
  - writing impostors, 192
- incremental builds, Eclipse, 163
- indentation settings
  - autoformatting, Pydev, 252
- info command, Subversion, 49
  - verifying status, 117
- info directory
  - configuring build slave, Buildbot, 113
- init (`__init__`) method
  - application initializing dependencies,
    - pMock, 211
  - creating FeedEntry factory, pMock, 201
  - mocking `__init__` directly, PyMock, 222

- install command, Setuptools, 90
  - ensuring local dependency processing, 132
  - installing from local copy, 96
- Install step, Buildbot, 125–128
  - ensuring local dependency processing, 134
- installations
  - Buildbot, 104–106
  - DBMigrate, 300
  - Eclipse, 25
  - Eclipse plug-ins, 30
  - eggs, 84
  - JUnit, 321
  - Mylyn, 25
  - pMock, 195
  - Pydev, 32
  - Setuptools, 86–87
  - SQLAlchemy, 283
  - SQLObject, 267
  - Subversion, 44–47
  - Twisted, 104
- InstallReports class, 362
- install\_lib command, Setuptools, 90
- install\_requires attribute, setup.py, 92
- integration testing
  - description, 140, 339
  - Framework for Integrated Tests (FIT), 340–353
    - running FIT with Buildbot, 353–367
- intermediateTable keyword, SQLObject, 282
- interpreter, Python
  - paths used on build slave, 124
- isolating components under test, 190–192
- isolation, databases, 264
- is\_empty method, pMock, 210
- is\_empty method, PyMock, 225, 226
- iterative design methods, 175
- iterative methodologies, 3

## J

- JavaScript, 312
  - getElementById method, 335
  - HTML documents and forms, 337
  - interacting with browser, 328
  - NaN value, 327
  - null value, 327

- pressCalculate method, 335
- setUp function, 330
- “Show all” errors window, 332
- suite function, 335
- testing, 320, 326
- undefined value, 327
- unit testing, 337
- validateSlope method, 331, 335
- validation function, 329
- variable declaration/assignment, 327
- jMock, 193
- job management system, 21
- join method, SQLAlchemy, 295
- join statements, SQLAlchemy, 279
- joinColumn keyword, SQLAlchemy, 282
- joins, SQLAlchemy, 275, 276
- JUnit, 321–326
  - adjusting timeouts, 324
  - aggregating tests, 335–336
  - author, 321
  - correspondence with unittest, 326
  - description, 337
  - distributed mode, 337
  - failures and errors, 331
  - installing, 321
  - package, 321
  - running tests by URL, 336
  - stand-alone testing, 322
  - test runner, 322
- Jython, 33
- K**
- keys
  - foreign keys, SQLAlchemy, 275
- keyword queries, SQLAlchemy, 290
- Komodo IDE, 22
- L**
- labels
  - revision control systems, 44
- landing page, Buildbot, 111
- libraries
  - mocking libraries, 193–194
- library directories, Pydev, 33
- license agreements
  - installing Eclipse plug-ins, 28
- linear independence
  - cyclomatic complexity, 239
- Linux
  - usermod command, 117
- list command, Subversion, 46
- listing method, pMock
  - FeedEntry factory, 203
- listing\_from\_item method, 176
- list\_from\_item method, 177
- load testing, 140
- local dependencies
  - ensuring local dependency processing, 132
- Location field
  - builder properties window, 165
  - External Tools dialog, Eclipse, 169
- log command, Subversion, 56
- low test coverage, 238
- M**
- MailNotifier class
  - configuring build master, Buildbot, 110
- mailto scheme, 311
- main method
  - converting URLs into feed objects, 188
  - hook between Setuptools and application class, 175
  - implementing application class, 176
  - new\_main method, pMock, 210, 212
  - test\_main test, pMock, 212
  - test\_main test, PyMock, 227
- Main tab, builder properties window, 165
- makefiles, Buildbot
  - configuring build master, 108
  - configuring build slave, 113
- many-to-many relationships
  - SQLAlchemy, 293
  - SQLObject, 277
- mapper directive
  - one-to-many relationships, SQLAlchemy, 292
- mapping layer database interactions, 297
- markup
  - MVC testing web applications, 317–320
- master *see* build master, Buildbot
- master.cfg file
  - configuring build master, Buildbot, 108
  - paths used on build slave, 124
  - running FIT with Buildbot, 353, 354
  - using source code, 119

- McCabe complexity *see* cyclomatic complexity
- measurements, 236–237
  - code coverage, 237–239
  - coding conventions, 244
  - complexity measurements, 239–242
  - development velocity, 242–243
  - effecting change, 236
  - factors characterizing, 236
  - feedback, 235, 236, 261
  - instrument for measuring, 236
  - low test coverage, 238
  - purpose of, 236
  - qualitative measurements, 235, 243–246, 261
  - quantitative measurements, 235, 237–243, 261
  - scope of, 236
  - side effects, 236–237
  - software quality, 235
  - units of measurement, 236
  - what is being measured, 236
- merge command, Subversion, 56
- merging files
  - revision control systems, 43–44
- MetaData class, SQLAlchemy, 284
- method function, PyMock, 215
- methodologies *see* development methodologies
- methods
  - agile methods, 4
  - decorators modifying, 161
  - how dependencies arise, 190
  - overriding existing methods, 185–189
  - protocols, 193
  - testing and, 142
- middleware, WSGI, 316
- migration dictionary, DBMigrate, 303
- migrations, database, 298–306
  - DBMigrate, 300–306
  - generating migration instructions, 299
  - locating migration mechanism, 300
  - numbering migrations and playback, 299
  - testing databases, 298
- mkdir command, Subversion, 46
- mock object packages
  - types (pMock/PyMock) compared, 231
- mock objects
  - attribute setter mocking, PyMock, 229
  - benefit of, 206
  - code coverage, 238
  - description, 192
  - domain-specific languages (DSL), 193
  - EasyMock, Java, 193
  - exception mocking, pMock, 228
  - exception mocking, PyMock, 228
  - generator mocking, PyMock, 230
  - introducing into tested code, pMock, 196
  - mocking class constructors, PyMock, 220
  - pMock, 194, 195–212
  - PyMock, 212–228
  - specifying calls on mock objects, PyMock, 215
  - switching from record mode to replay mode, PyMock, 213
  - verify method, PyMock, 213
- mocking
  - breaking dependencies, 191
- mocking libraries, 193–194
- model-view-controller (MVC), 316, 337
- modules
  - mocking external modules, PyMock, 216
- modules, Pydev, 33
- modules, Python, 82
  - coupling, 141
- mod\_python, Apache plug-in, 314
- monkeypatching, 185–189
  - description, 231
  - drawbacks if hand coding, 192
  - imports and, 186
  - mocking calls to self, pMock, 196
  - pMock, 193
  - PyMock, 214–215
    - moving methods to new object, 218
    - objects that Python can't monkeypatch, 224
  - writing impostors, 192
- motivation
  - rewarding good code, 248–249
- move command, Subversion, 51
- moving files, Subversion
  - working from command line, 51
  - working through Eclipse, 77
- multiple joins, SQLAlchemy, 276

multiple relationships, SQLAlchemy, 280

MVC (model-view-controller)

testing web applications, 316, 337

Mylyn

advantages of Eclipse IDE, 21

installing, 25

job management system, 22

## N

name attribute, setup.py, 87

naming conventions/standards

camel case, 248

coding conventions, 244

renaming, 183–184, 250

typographical naming conventions, 248

NaN value, JavaScript, 327

native tests, Nose, 149

never calling policy, pMock, 229

new\_main method, pMock, 210, 212

new\_main method, PyMock, 226

Nightly scheduler, 366

Node Kind field

info command, Subversion, 49

nodes, HTML, 310

Nose

attributes, 160, 174

coverage package, 258

description, 149, 174

discovering unit tests, 149

finding tests with Nose, 159–160

implementing application tests as Nose  
tests, 176

introduction, 148

native tests, 149

options in builder properties window, 165

running coverage through Nose, 258

running full test suite in development, 167

skipping slow tests, 160–162

stdout and stderr, 160

tags, 160

tests\_require property, 167

tests\_suite property, 167

turning off output capture, 160

underscore (\_) prefix, 195

nosetests, 159

builder properties window, 165

negating options, 161

standard test output, 160

vociferous test output, 160

with-coverage option, 258

notification classes

configuring build master, Buildbot, 110

notifications, 121

null value, JavaScript, 327

nullable keyword, SQLAlchemy, 285

## O

-o option, unzip command, 360–361

object-relational mappers *see* ORMs

object-relational mismatch, 264

on-site customers, 8–9

once calling policy, pMock, 229

once playback count, PyMock, 229

one method, SQLAlchemy, 288

one-to-many relationships

SQLAlchemy, 291

SQLObject, 275

one\_or\_more playback count, PyMock, 229

operator overloading, Python, 193

organizational authority, 245

ORMs (object-relational mappers), 265–267

active record pattern, 266

aspects of ORMs, 265

data mapper pattern, 266

description, 306

Python ORMs, 267–296

SQLAlchemy, 283–296

SQLObject, 267–282

unit of work pattern, 266

otherColumn keyword, SQLAlchemy, 282

overloading operators, Python, 193

override expression, PyMock, 215

override function, PyMock, 215–216

monkeypatching, 214

overriding objects

monkeypatching and imports, 186

## P

packages

catching missing packages, 132

coverage package, 258

FeedParser package, 149

Install step, Buildbot, 125

paths used on build slave, 124

packages directive, Setuptools, 88

packages, Pydev, 33

- packages, Python, 82
  - egg naming structure, 84
  - installing, 83–84
  - managing dependencies, 83
  - packages for unit testing, 148
- packages, Setuptools
  - finding packages on the Net, 94
  - managing dependencies, 92–94
  - removing existing package, 95
- package\_dir directive, Setuptools, 88
- package\_reports.py tool, 358
- pair programming, 5–7, 249
- params dictionary
  - running DBMigrate from program, 306
- parsing
  - FeedParser package, 149
  - HTMLParser library, 318
  - MVC testing markup, 318
- path manipulation solution
  - replicable builds, 85
- paths, Buildbot
  - absolute paths, 124
  - paths used on build slave, 124
  - relative paths, 124
- Pattern field, Pydev templates, 255
- PBChangeSource class
  - configuring build master, Buildbot, 109
- pending prefix
  - AggregateFeed creating FeedEntry factory, pMock, 200
  - testing, 195
- PEPs (Python Enhancement Proposals), 247
  - PEP 20 - The Zen of Python, 247
  - PEP 257 - Docstring Conventions, 248
  - PEP 333 - Web Server Gateway Interface, 314
  - PEP 8 - Style Guide for Python Code, 248
- performance testing, 140, 339
- perspectives, Eclipse, 24
- phytoplankton host, aliasing, 104
- planning
  - agile development and, 4
- playback counts, pMock/PyMock, 229
- plug-ins
  - installing Eclipse plug-ins, 30
- pMock, 195–212
  - AggregateFeed creating FeedEntry factory, 200
  - AggregateFeed initializing FeedParser factory, 203
  - AggregateFeed.entries initialized to Set, 201
  - application initializing dependencies, 211
  - creating aggregate entries for feeds, 197
  - description, 193–194
  - empty AggregateFeed output, 209
  - exception mocking, 228
  - expects clause, 196
  - FeedWriter initializing stdout attribute, 209
  - formatting feed entry listings, 207
  - installing, 195
  - introducing mocks into tested code, 196
  - mock object packages compared, 231
  - mocking calls to self, 196
  - modeling basis for, 175, 231
  - playback counts, 229
  - raising exceptions with pMock, 228
  - refactoring/condensing tests, 206
  - refactoring/extracting AggregateFeed, 198
  - same constraint, 196
  - taking and converting feeds, 200
  - test\_add, 201
  - test\_add\_single\_feed, 197, 199
  - test\_aggregate\_feed\_creates\_factory, 206
  - test\_aggregate\_feed\_initializes\_feed\_parser test, 203, 206
  - test\_combine\_feeds, 196
  - test\_create\_entry, 200
  - test\_entries\_is\_always\_defined, 201
  - test\_feed\_entry\_from\_parsed\_feed, 208
  - test\_feed\_entry\_listing, 202
  - test\_feed\_writer\_initializes\_stdout, 209
  - test\_feed\_writer\_prints\_nothing\_with\_an\_empty\_feed, 209
  - test\_from\_urls, 204
  - test\_get\_feeds\_from\_urls, 203
  - test\_is\_empty, 210
  - test\_main, 212
  - test\_new\_main, 210
  - test\_print\_entry\_listings, 208, 210
  - test\_rreader\_initializes\_dependencies, 211
  - turning new\_main into main method, 212
  - verifying dependencies initialized correctly, 206

- verifying FeedEntry factory operation, 202
- will clause, 196
- pMock methods
  - add method, 201
  - combine\_feeds method, 199
  - create\_entry method, 200
  - defining feeds\_from\_urls method, 203
  - feeds\_from\_urls method, 204
  - feed\_entry\_listing method, 202
  - from\_parsed\_feed method, 202, 208
  - from\_urls method, 204
  - is\_empty method, 210
  - new\_main method, 210, 212
  - print\_entry\_listings method, 208, 210
  - verify method, 196
- port keyword, DBMigrate, 301
- post-commit hook, Buildbot, 121
- Postel's Law, 311
- pre-commit hook, Buildbot, 121
- pre-commit hooks, 256–257
- pressCalculate method, JavaScript, 335
- primary keys, SQLAlchemy, 284
- printed\_items value, 179
  - breaking into strings, 182
  - renaming, 183
- print\_entry\_listings method
  - pMock, 208, 210
  - PyMock, 224, 225
- processConnection variable
  - setting up SQLite connection, 269
- production database
  - using for development, 297
- programmer tests, 139
  - see also* unit tests
- programming
  - cohesion, 141
  - collective code ownership, 12–13
  - coupling, 141
  - pair programming, 249
  - web application frameworks, 337
- project file, Eclipse, 38
- project major/minor, 45
- project properties window, 163
- projectName property, Buildbot, 110, 112
- projects
  - building projects with setup.py, 88–91
  - Import project window, 60
  - job management system, 21
  - organizing projects, 45
  - shared projects, 59
  - starting new Pydev project, 32–38
- projectUrl property, Buildbot, 110, 112
- properties, Subversion
  - ignoring files, 99
- propset command, Subversion, 99
- protocols, Python, 193
- protocols, Web, 312
- public\_html directory, Buildbot, 108
- pw keyword, DBMigrate, 301
- PyChecker, 256
- Pydev
  - autoformatting, 251
  - creating Python class, 33
  - Eclipse, 25
  - Eclipse plug-ins, 22
  - features assisting writing/analyzing code, 251
  - installing, 32
  - library directories, 33
  - modules, 33
  - packages, 33
  - renaming unit test values for readability, 183
  - starting new project, 32–38
  - templates, 254
- Pydev extensions, 39
  - code analysis features, 253
  - Eclipse plug-ins, 22
- Pydev Package Explorer pane, 33
  - working with Subversion through Eclipse, 68
- pydevproject file, Eclipse, 38
- pydistutils.cfg file, 85, 86
- PyFit, 340–353
  - creating column fixture test with FIT, 346–353
  - running PyFit, 349–353
  - creating directory for acceptance tests, 346
  - description, 368
  - example, 344–345
  - FitNesse, 368
  - fixtures, 348
  - programs for running tests, 349



- requirement documents, 341–344
- running FIT with Buildbot, 357
  - making reports available, 358
- spreadsheet support, 341
- PyLint, 256
- PyMock, 212–228
  - activating new functionality, 227
  - AggregateFeed.entries initialized to Set, 222
  - AggregateFeed.\_\_init\_\_, 221
  - alternative syntax to define expectations, 214
  - attribute setter mocking, 229
  - dummy objects, 216
  - ending keyword, 230
  - entry listings sorting test, 223
  - eq constraint, 213
  - exception mocking, 228
  - expects clause, 215
  - FeedEntry.\_\_init\_\_, 222
  - generator mocking, 230
  - getitem (\_\_getitem\_\_) function, 213
  - method function, 215
  - mock object packages compared, 231
  - mocking class constructors, 220
  - mocking external modules, 216
  - mocking \_\_init\_\_ directly, 222
  - modeling basis for, 175, 231
  - monkeypatching, 214–215
  - moving methods to new object, 218
  - objects that Python can't monkeypatch, 224
  - override function, 215–216
  - playback counts, 229
  - raising exceptions with PyMock, 228
  - record-replay model, 213
  - replay mode, 213
  - returns clause, 215
  - RSReader initialization test, 227
  - specifying calls on mock objects, 215
  - switching from record mode to replay mode, 213
  - test\_add, 221
  - test\_add\_single\_feed, 217
  - test\_entries\_is\_always\_defined, 221
  - test\_feed\_entry\_constructor, 222–223
  - test\_feed\_entry\_listing, 222
  - test\_from\_urls, 216, 219
  - test\_getting\_attributes, 229
  - test\_is\_empty, 226
  - test\_main, 227
  - test\_new\_main, 226
  - test\_print\_agg\_feed\_listing\_is\_printed, 224
  - test\_rsreader\_dependency\_initialization, 227
  - test\_setting\_attributes, 229
  - use\_pymock decorator, 212
  - using PyMock with unittest, 230
- PyMock methods
  - add method, 221
  - add\_single\_feed method, 216–217, 218
  - create\_entry method, 217–218, 220–221
  - entry\_listings method, 223
  - from\_urls method, 215–216, 219
  - print\_entry\_listings method, 224, 225
  - verify method, 213
- PyTest, 149
- Python
  - build slaves and, 122
  - cheese shop repository, 84
  - coding standards, 246
  - creating classes, 33
  - decorators, 161
  - egg naming structure, 84
  - ez\_setup.py program, 86
  - global entities, 191
  - modules, 82
  - ORMs, 267–296
  - package root, 88
  - packages, 82
    - installing, 83–84
    - managing dependencies, 83
    - unit testing, 148
  - protocols, 193
  - site package mechanism, 86
  - types of defective coding, 251
  - typographical naming conventions, 248
- Python 2.4
  - supporting Python 2.4 builds, 128–132
- Python application servers, 313
- Python development environment
  - creating for Eclipse IDE, 40
- Python Enhancement Proposals *see* PEPs

- Python interpreter, 124
- Python Interpreters screen, 32
- PYTHONPATH variable, 32
  - installing eggs, 84
  - replicable builds, 85
  - running PyFit, 350–351
  - specifying additional packages, 82

## Q

- qq option, unzip command, 360
- quality, software, 234
- qualitative measurements, 243–246
  - coding conventions, 244
  - description, 235, 261
- quantitative measurements, 237–243
  - code coverage, 237–239
  - complexity measurements, 239–242
  - description, 235, 261
  - development velocity, 242–243
- queries, SQLAlchemy, 287
  - chained expressions, 290
  - filtering with SQL queries, 289
  - keyword queries, 290
  - querying relations, 295
- query method, SQLAlchemy, 287
  - chained expressions, 290

## R

- raises method, PyMock, 228
- raise\_exception function, pMock/PyMock, 228
- readability
  - renaming, 183
- reconfig command, Buildbot, 119
- reconfiguration, Buildbot, 120
- record mode, PyMock, 213
- record-replay, EasyMock, 193
- record-replay model, pMock, 194
- record-replay model, PyMock, 213
- refactoring, 11–12
  - add\_single\_feed method, PyMock, 218
  - condensing tests, pMock, 206
  - creating rows, SQLAlchemy, 271
  - databases, 264, 298
  - extracting AggregateFeed, pMock, 198
  - implementing application class, 176
  - moving from\_urls, PyMock, 219
  - moving methods to new object, PyMock, 218
  - reasons for, 143
  - reimplementing from\_urls, pMock, 204
  - running unit tests manually in Eclipse, 155–156
  - triangulation, 182
  - turning new\_main into main method, pMock, 212
- refactoring menu, Pydev renaming capability, 183
- refactoring preview window, Pydev, 184
- refactoring tools, 250
- references to objects
  - monkeypatching and imports, 186–187
- regression testing, 10, 140
- relationships
  - many-to-many, SQLAlchemy, 293
  - many-to-many, SQLAlchemy, 277
  - multiple, SQLAlchemy, 280
  - one-to-many, SQLAlchemy, 291
  - one-to-many, SQLAlchemy, 275
  - querying, SQLAlchemy, 295
- relative paths, Buildbot, 124
- relocate option, switch command, 118
- relying on the compiler technique, 178
- Rename refactoring window, Pydev, 184
- renaming, 183–184, 250
  - working with Subversion through Eclipse, 77
- replay mode, PyMock, 213
- replicable builds, 85–86
- reports
  - Framework for Integrated Tests (FIT), 341
- running FIT with Buildbot, 358–366
  - getting revision, 361–362
  - making reports available, 358
  - packaging reports, 358–359
  - publishing build, 362–366
  - publishing reports, 360–366
  - retrieving reports, 359–360
- repository
  - continuous integration, 16
- repository, Python
  - cheese shop, 84

- repository, Subversion
    - accessing, 45
    - creating, 44
    - defining repository location, 61
    - repository tree, 116
    - team repository view, 65–68
  - requests, HTTP, 312
  - requirement documents
    - design and testing, 368
    - Framework for Integrated Tests (FIT), 340–341, 346
    - writing requirements, 341–344
    - writing HTML spec document, 342
  - resolved command, Subversion, 58
  - responses, HTTP, 312
  - results, SQLAlchemy, 287
  - return values
    - how dependencies arise, 191
  - returns clause, PyMock
    - defining expectations, 215
  - revert command, Subversion, 53
  - Revert window
    - working with Subversion through Eclipse, 73
  - reviews
    - code reviews, 249
    - short iterations, 13
  - revision control systems
    - benefits of, 41
    - benefits of Subversion, 42–44
    - branches, 44
    - distributed revision control systems, 42
    - edit-and-merge process, 43
    - exclusive locking, 43
    - labels, 44
    - merging files, 43, 44
    - subverting Eclipse, 59–64
    - types of, 42
    - working with Subversion from command line, 47–59
    - working with Subversion through Eclipse, 64–79
  - Revision field
    - info command, Subversion, 49
  - revision flag, DBMigrate, 304
  - revision property
    - publishing reports, 361
  - revisions keyword
    - running DBMigrate from program, 306
  - RFC 2616, 312
  - RFC 3875, 313
  - RFC 822, 312
  - rows, SQLAlchemy
    - creating, 269–272
    - deleting, 275
  - RSReader application
    - acceptance tests, 152
    - application initializing dependencies, pMock, 211
    - building project, 88–91
    - building RSS reader, 81
    - converting URLs into feed objects, 185, 188
    - finding packages, 88
    - identifying project with name attribute, 87
    - implementing application class, 176
    - initialization, PyMock, 227
    - installing executables using setup.py, 91
    - installing rsreader into site-packages, 89
    - managing dependencies, 92–94
    - test\_rsreader\_dependency\_initialization, PyMock, 227
    - test\_rsreader\_initializes\_dependencies, pMock, 211
    - using data files, 189
    - with unit test skeleton, 152
  - rsreader directory
    - paths used on build slave, 124
  - RSReader links, Buildbot, 112
  - rsreader-linux slave, Buildbot, 108
  - RSS feeds, 149
  - RSS reader application
    - acceptance testing, 150
    - aggregators and, 149
    - building simple RSS reader, 81
  - running commands, Setuptools, 89
- S**
- same constraint, pMock, 196
  - Save as field
    - Common tab, External Tools dialog, 170
  - scheduler
    - running FIT with Buildbot, 353, 354

- schedulers property, Buildbot
  - configuring build master, 109
  - supporting Python 2.4 builds, 130
- schema definition, SQLAlchemy, 284
- schema definition, SQLAlchemy, 272
- scheme keyword, DBMigrate, 301
- scrum methodology, 243
- secondary keyword, SQLAlchemy, 294
- select method, SQLAlchemy, 273
- selectBy method, SQLAlchemy, 274
- self
  - mocking calls to self, pMock, 196
- semantic verifiers, 256
- sendchange command, Buildbot, 114, 120
  - paths used on build slave, 125
- servers
  - build servers, 103
  - startup script, 112
- sessionmaker function, SQLAlchemy, 285
- sessions, SQLAlchemy, 285–286
- setProperty method, BuildStep class, 361
- setter mocking, PyMock, 229
- setUp function, JavaScript, 330
- setUp method
  - correspondence of unittest/JsUnit, 326
  - running unit tests manually in Eclipse, 155
  - using PyMock with unittest, 230
- setup.cfg file
  - fixing options with setup.cfg, 97
  - Subversion ignoring files, 98
- setup.py program, Setuptools, 87
  - bootstrapping Setuptools, 97
  - building packages, 83
  - building projects, 88–91
  - console\_scripts key, 91
  - entry\_points attribute, 91
  - external\_requirements attribute, 92
  - gui\_scripts key, 91
  - installing executables, 91
  - install\_requires attribute, 92
  - name attribute, 87
  - version attribute, 87
- Setuptools, 81
  - basic usage, 87
  - bdist\_egg command, 90
  - build command, 89
  - building projects with setup.py, 89
  - build\_ext module, 89
  - build\_py command, 89, 90
  - configuring values for options, 97
  - description, 84
  - development mode, 100–102
  - dist directory, 91
  - Distutils compared, 102
  - easy\_install program, 87
  - eggs, 81
  - egg\_info command, 90
  - finding packages on the Net, 94
  - find\_packages function, 88
  - hook to application class, 175
  - install command, 90
  - installing, 86–87
  - installing eggs, 84
  - installing executables using setup.py, 91
  - install\_lib command, 90
  - managing dependencies, 92–94
  - packages directive, 88
  - package\_dir directive, 88
  - removing existing package, 95
  - running commands, 89
  - running full test suite in development, 167
  - setup.py program, 87
  - Subversion ignoring files, 98–100
  - version numbers, 88
  - zip\_safe flag not set warning, 94
- set\_count playback count, PyMock, 229
- SGML (Standard Generalized Markup Language)
  - HTML and, 310
  - XML and, 311
- shared projects, 59
- ShellCommand build
  - Buildbot running full test suite, 171
- ShellCommand classes, 361
- short iterations, 13–15
- “Show all” errors window, JavaScript, 332
- simple design, 12
- site package mechanism, Python, 86
- site-packages directory, Python, 82
  - installing Python packages, 83
  - installing rsreader into site-packages, 89
  - replicable builds, 85
- site-packages directory, Setuptools
  - development mode, 100
  - removing existing package, 95

- slave *see* build slave, Buildbot
- slave-lnx01
  - aliasing hosts, 104
  - configuring build slave, Buildbot, 112
  - configuring build system, 106
- slavePortnum property, Buildbot, 108, 109
- slaves property, Buildbot
  - configuring build master, 108
  - configuring build slave, 112
- socket keyword, DBMigrate, 301
- software quality, 234
  - measurements, 235
- source
  - aliasing hosts, 104
  - naming Subversion for use with Buildbot, 104
- source code
  - build slave obtaining, 119
- source control
  - build master/slave accessing Subversion, 116–118
- source folders, Eclipse, 33
- source repository, 40
- SQL injection attacks, 290
- SQLAlchemy, 283–296
  - all method, 288
  - backref keyword, 294
  - cascade keyword, 295
  - chained expressions, 290
  - connecting to database, 283
  - delete method, 295
  - filter method, 289
  - filter\_by method, 290
  - first method, 288
  - from\_statement method, 290
  - further documentation on, 296
  - installing, 283
  - join method, 295
  - keyword queries, 290
  - many-to-many relationships, 293
  - mapper directive, 292
  - MetaData class, 284
  - nullable keyword, 285
  - one method, 288
  - one-to-many relationships, 291
  - primary keys, 284
  - query method, 287
    - chained expressions, 290
    - querying relations, 295
  - results, 287
  - secondary keyword, 294
  - sessionmaker function, 285
  - sessions, 285–286
- SQLBuilder
  - retrieving objects, SQLAlchemy, 273
- SQLExplorer, Eclipse plug-ins, 22
- SQLite connection, setting up, 269
- SQLObject, 267–282
  - attribute defaults, 272
  - connecting to database, 268, 269
  - creating rows, 269–272
  - deleting rows, 275
  - describing database table to, 268
  - foreign keys, 275
  - installing, 267
  - join statements, 279
  - many-to-many relationships, 277
  - multiple joins, 276
  - multiple relationships, 280
  - one-to-many relationships, 275
  - retrieving objects, 273
  - schema definition, 272
  - updating fields, 274
- src directory, Eclipse, 39
- SSH trust relationships, 118
- Standard Input and Output field
  - Common tab, External Tools dialog, 170
- start-commit hook, Buildbot, 121
- startup screen, Eclipse, 24
- startup script, servers, 112
- start\_response() callback, 315
- state
  - HTTP protocol, 312
- statement coverage, 237
- status command, Subversion, 49, 50, 53, 54, 56
  - ignoring files, 98
- status property, Buildbot, 110
- stderr
  - Nose intercepting, 160
- stdin, HTTP request, 313
- stdout
  - defining print\_entry\_listings, pMock, 208
  - HTTP response, 313
  - Nose intercepting, 160

- running unit tests manually in Eclipse, 153, 158
- stdout attribute, pMock, 209
- steps, Buildbot *see* build steps, Buildbot
- stories, 339
- stress testing, 140
- string expansion
  - creating migrations, DBMigrate, 302
- StringIO object
  - running unit tests manually in Eclipse, 153
- stubs, 192
- subclassing, 193
- Subversion, 41
  - accessing repositories, 45
  - add command, 50
  - benefits of, 42–44, 79
  - checkout command, 47, 61
  - commit command, 50, 121
  - copy command, 51, 78
  - creating repository, 44
  - delete command, 52, 77
  - diff command, 54–56
  - directory structure, 44
  - events and hooks, 121
  - ignore property, 99
  - ignoring files, 98–100
  - import command, 46
  - importing into Eclipse from, 60–64
  - info command, 49
  - installing, 44–47
  - list command, 46
  - log command, 56
  - merge command, 56
  - mkdir command, 46
  - move command, 51
  - organizing projects, 45
  - parts of Subversion URL, 45
  - propset command, 99
  - resolved command, 58
  - revert command, 53
  - sharing subverted project, 59–60
  - status command, 49, 50, 53, 54, 56
  - svn directories, 43
  - update command, 54, 56
  - validating committed code, 256–257
  - working from command line, 47–59
    - adding files, 50–51
    - conflicting file changes, 55–59
    - copying/moving files, 51–52
    - deleting files, 52
    - examining workspace files/directories, 49–50
    - merging code, 55–59
    - modifying files, 53–54
    - restoring files, 53
    - updating working copy, 54–55
  - working through Eclipse, 64–79
    - adding files, 68–69
    - committing changes, 70–71
    - copying files, 78–79
    - deleting files, 76–77
    - editing files, 71–72
    - moving files, 77
    - renaming files, 77
    - resolving conflicts, 73–76
    - reverting changes, 72–73
    - reverting moves/renames/copies, 79
    - team repository view, 65–68
- Subversion clients
  - accessing local filesystem, 117
  - accessing svnserve, 117
- Subversion group
  - adding committers to, 117
  - permissions, 116
- Subversion repository
  - access methods, 116
  - authorization/permissions, 116
  - baseURL property, 119
  - build master/slave accessing, 116–118
  - changing permission ownership, 117
  - defaultBranch property, 119
  - naming for use with Buildbot, 104
  - repository tree, 116
- Subversive plug-in, 22
  - Subversion revision control systems, 41
  - team providers, 59
- suite function, JavaScript
  - aggregating JsUnit tests, 335, 336
- SVN build step
  - paths used on build slave, 124
  - publishing reports, 361
  - using source code, 119
- svn directories, 43

- svn group
    - Subversion repository, 116
    - switch command, 118
  - SVN step, Buildbot, 119–121
  - svn user, creating, 117
  - Svnhooks, 256
  - svnlook
    - precommit hooks, 256
  - svnserve
    - build master/slave accessing Subversion, 116
    - committers and file ownership, 116
    - daemon option, 117
    - starting after host machine reboot, 118
    - Subversion permissions, 116
  - switch command, svn, 118
  - Synchronize view
    - working with Subversion through Eclipse, 65, 68
  - sys.argv file, 154
  - sys.stdout file, 153, 158
  - system metaphor, 8
- T**
- Table method, SQLAlchemy, 284
  - tables
    - creating tables, SQLAlchemy, 270
    - many-to-many relationships, SQLAlchemy, 293
    - many-to-many relationships, SQLAlchemy, 277
    - multiple relationships, SQLAlchemy, 280
    - one-to-many relationships, SQLAlchemy, 291
    - one-to-many relationships, SQLAlchemy, 275
  - tags, HTML, 310
  - tags, Nose, 160
    - tag expressions, 162
  - tasks
    - job management system, 22
  - TDD (test-driven development), 10–11, 146–147
    - description, 173
    - implicit goal, 218
    - using mock objects
      - pMock, 195–212
      - PyMock, 212–228
  - team providers, 59
  - team repository view, 65–68
  - tearDown method
    - correspondence of unittest/JsUnit, 326
    - running unit tests manually in Eclipse, 155
    - using PyMock with unittest, 230
  - technical authority, 245
  - templates, Pydev, 254
  - test assertion
    - implementing application tests, 176
  - test coverage *see* code coverage
  - test doubles *see* impostors
  - test fixtures, 142
  - test from\_urls test, pMock, 204
  - test harness
    - converting URLs into feed objects, 185
    - Setuptools, 81, 102
  - test methods/functions
    - correspondence of unittest/JsUnit, 326
  - test package
    - running unit tests manually in Eclipse, 152
  - test runners
    - Framework for Integrated Tests (FIT), 341
    - JsUnit, 322
  - test suite pages
    - aggregating JsUnit tests, 335–336
  - test-driven development *see* TDD
  - TestCase classes
    - Nose, 149, 159
    - unittest, 148
  - testGeneratesWithRaisedTermination, PyMock, 230
  - testing
    - acceptance testing, 140, 339–353
    - application tests, creating, 175
    - customer tests, 139
    - databases, 264
    - exploratory testing, 140
    - Framework for Integrated Tests (FIT), 340–353, 367
    - functional testing, 139, 339
    - integration testing, 140, 339
    - isolating components, 190–192
    - JavaScript, 320–326
    - load testing, 140
    - pending prefix, 195
    - performance testing, 140, 339



- programmer tests, 139
  - PyFit, 340–353
  - refactoring/condensing tests, pMock, 206
  - regression testing, 10, 140
  - running tests by URL, 336
  - running tests manually in Eclipse, 151–159
  - stories, 339
  - stress testing, 140
  - unit testing, 9–10, 141–142
  - using data files, 189
  - web applications, 316
    - graphics and images, 317
    - markup, 317
    - MVC, 316
  - XUnit test pattern, 332
  - testPage parameter, JsUnit
    - running tests by URL, 337
  - TestRunner objects, unittest, 148
  - TestSuite classes, unittest, 148
  - tests\_require property, Nose, 167
  - test\_add, pMock, 201
  - test\_add, PyMock, 221
  - test\_add\_single\_feed, pMock, 199
  - test\_add\_single\_feed, PyMock, 217
  - test\_aggregate\_feed\_creates\_factory, pMock, 206
  - test\_aggregate\_feed\_initializes\_feed\_parser, pMock, 203, 206
  - test\_combine\_feeds, pMock, 196
  - test\_create\_entry, pMock, 200
  - test\_entries\_is\_always\_defined, pMock, 201
  - test\_entries\_is\_always\_defined, PyMock, 221
  - test\_feed\_entry\_constructor, PyMock, 222–223
  - test\_feed\_entry\_from\_parsed\_feed, pMock, 208
  - test\_feed\_entry\_listing, pMock, 202
  - test\_feed\_entry\_listing, PyMock, 222
  - test\_feed\_listing, 181–182
  - test\_feed\_writer\_initializes\_stdout, pMock, 209
  - test\_feed\_writer\_prints\_nothing\_with\_an\_empty\_feed, pMock, 209
  - test\_from\_urls, PyMock, 216, 219
  - test\_getting\_attributes, PyMock, 229
  - test\_get\_feeds\_from\_urls, pMock, 203
  - test\_is\_empty, pMock, 210
  - test\_is\_empty, PyMock, 226
  - test\_listing\_for\_item function, 182
  - test\_listing\_from\_item function, 183
  - test\_main, pMock, 212
  - test\_main, PyMock, 227
  - test\_new\_main, pMock, 210
  - test\_new\_main, PyMock, 226
  - test\_print\_agg\_feed\_listing\_is\_printed, PyMock, 224
  - test\_print\_entry\_listings, pMock, 208, 210
  - test\_rsreader\_dependency\_initialization, PyMock, 227
  - test\_rsreader\_initializes\_dependencies, pMock, 211
  - test\_setting\_attributes, PyMock, 229
  - test\_suite property, Nose, 167
  - triangulation, 182
  - Twisted, 104, 364
  - typing
    - duck typing, 193
  - typographical naming conventions, Python, 248
- ## U
- umask command
    - Subversion group permissions, 116
  - undefined value, JavaScript, 327
  - underscore (\_) prefix, Nose, 195
  - unit of work pattern, ORMs, 266
  - unit tests, 9, 141–142
    - arguments against unit testing, 143–146
      - don't understand code behavior, 144
      - environment for running tests, 144
      - not developer's job, 144
      - time spent on unit testing, 143, 144
      - unreliable tests, 144
    - asserting success or failure, 147
    - cohesion and coupling, 141
    - collective code ownership, 143
    - description, 139, 173
    - disabling unit tests, 145
    - discovering unit tests, 149
    - extent, 145
    - feedback on defective code, 251
    - finding tests with Nose, 159–160
    - fixing broken tests, 145
    - how they work, 141
    - isolating components, 190–192

- JavaScript, 337
  - locating test code, 148
  - observing failing test first, 152
  - preventing regression, 145
  - problems with not unit testing, 142–143
  - Python packages for, 148
  - renaming values for readability, 183
  - RSReader with unit test skeleton, 152
  - running DBMigrate with unit tests, 305
  - running full test suite at build time, 171–173
  - running full test suite in development, 167–171
  - running tests after every change, 163–166
  - running tests by URL, 336
  - running tests manually in Eclipse, 151–159
  - size of unit test suite, 353
  - structure, 174
  - test-driven development, 146–147
  - test fixtures, 142
  - testing JavaScript, 320–326
  - tools producing unit tests from finished code, 145
  - web application frameworks, 337
  - when unit tests need to be run, 162
  - why unit testing fails, 144
  - writing, 10
  - unittest
    - correspondence with JsUnit, 326
    - description, 148, 174
    - introduction, 148
    - TestCase classes, 148
    - TestRunner objects, 148
    - TestSuite classes, 148
    - using PyMock with unittest, 230
  - unzip command
    - publishing reports, 360
  - update command, Subversion, 54, 56
  - updating fields, SQLAlchemy, 274
  - upload directories
    - retrieving reports, 360
  - URLs (Universal Resource Identifiers)
    - options query, 311
    - schemes identifying resource type, 311
    - secondary index, 311
    - syntax and URLs, 311
    - Web and, 309, 337
    - when URI becomes URL, 312
  - URLs (Universal Resource Locators)
    - converting URLs into feed objects, 185, 188
    - feeds\_from\_urls method, pMock, 203
    - from\_urls method, PyMock, 216
    - parts of Subversion URL, 45
    - reimplementing from\_urls method, pMock, 204
    - running tests by URL, 336
    - steps followed by Web browsers, 310
    - test from\_urls, pMock, 204
    - URI syntax, 311
    - Web and, 309, 337
    - when URI becomes URL, 312
  - user keyword, DBMigrate, 301
  - user stories, 7–8
  - usermod command, Linux, 117
  - use\_pymock decorator, PyMock, 212
- ## V
- validateSlope function, JavaScript, 331, 335
  - validating committed code, 256–257
  - validation function, JavaScript, 329
  - velocity
    - development velocity, 242–243
  - verbose flag, DBMigrate, 304
  - verify method, pMock, 196
  - verify method, PyMock, 213
  - version attribute, setup.py, 87
  - version control
    - see also* revision control systems
    - benefits of, 41
  - versions
    - egg naming structure, 84
    - replicable builds, 85–86
    - version numbers, 88
  - versionable keyword, DBMigrate, 301
  - views, Eclipse, 24
  - virtual Python solution
    - replicable builds, 85
  - vocabulary
    - writing HTML spec document, 342
- ## W
- waterfall display, Buildbot, 111
    - description, 110
    - enhancing step progress description, 134

- illustration of, 111, 114, 119, 122, 125, 128, 136
  - running FIT with Buildbot, 355
  - waterfall methodology, 2
  - Web, 309, 337
  - web application frameworks, 313, 337
  - web applications, 312–314
    - testing, 316–320
      - graphics and images, 317
      - markup, 317
      - MVC, 316
    - using write() callback, 315
    - variations on implementation, 312
  - Web Server Gateway Interface (WSGI), 314–315
    - WSGI middleware, 316
  - web browsers
    - description, 310
    - JavaScript and, 312
    - JavaScript interaction with, 328
    - steps followed by, 310
  - Web Server Gateway Interface *see* WSGI
  - web servers
    - application framework connections, 313
    - using write() callback, 315
    - web applications and, 312
  - WebStatus class
    - configuring build master, Buildbot, 110
  - will clause, pMock, 196
  - Wing IDE, 22
  - with-coverage option, nosetests, 258
  - WithProperties class
    - publishing reports, 361
  - with\_sqlobject method
    - creating rows, SQLAlchemy, 271
    - schema definition, SQLAlchemy, 272
  - workbench, Eclipse, 25
  - working copy
    - updating working copy of file, 54–55
  - Working Directory field
    - builder properties window, 165
    - External Tools dialog, Eclipse, 169
  - World Wide Web, 309, 337
  - write() callback, 315
  - WSGI (Web Server Gateway Interface), 314–315
    - pronunciation of WSGI, 314
    - web application frameworks, 337
  - WSGI middleware, 316
- ## X
- xkcd\_output
    - renaming printed\_items value, 183
  - XML (eXtensible Markup Language), 311
    - SGML and, 310
  - XPath, 319
  - XUnit test pattern, 332
- ## Z
- zero\_or\_more playback count, PyMock, 229
  - zip\_safe flag not set warning, Setuptools, 94
  - Zope Interface, 104