

Index

Symbols

- `__add__` method 231
- `__class__` method 332
- `__doc__` method 74
- `__eq__` method 230
- `__getattr__` method 223
- `__getattribute__` method 225–227
- `__init__` method 185
- `__iter__` method 234
- `__lt__` method 231
- `__main__` method 108
- `__name__` method 108
- `__repr__` method 188
- `__setattr__` method 224
- `.center()` method 117
- `.endswith()` method 242
- `.groups()` method 243
- `.isdigit()` method 136
- `.ljust()` method 117
- `.py()` method 13
- `.rjust()` method 117
- `.split()` method 242
- `.write()` method 83
- `.writelines()` method 83
- `**` argument 113
- `**kwargs` 222
- `\n` 83
- `+` operator 96
- `+=` operator 126

A

- abstract class 208
- abstract data type 182
- adventure game
 - adding caves 199
 - combat 212
 - inventory 193
 - monsters 205
 - movement 201
 - verbs and nouns 189
- alpha value 316
- angles
 - acos and asin 304
 - and vectors 298–301
 - degrees vs. radians 300
- `apache_log_headers` 245
- architecture 100
- assert statement 103
- attribute
 - access, customizing 228
 - missing, looking up 223
 - setting 223

B

- backslash, creating special character with 81
- Beautiful Soup 148
 - `.contents` attribute 155
 - `.string` method 155
 - `find()` method 153
 - parsing HTML 152

- Bitbucket 387
- break statement 34, 44
- bug 39
- C**
- change monitor 97
- chat server 321–325
- ChatFactory 324
- checksum, generating 87
- choice function 38
- class 182–186
 - __init__ method 185
 - __repr__ method 188
 - abstract class 208
 - customizing 223–232
 - designing 194
 - Factory 324
 - inheritance 206
 - instances of 182
 - methods 185
 - mixin class 219–222
 - responsibility 193
 - subclasses 206
 - syntax 185
- collision detection 305
- Combined Log Format 241
- command, running 110–113
- command-line argument 77
- command-line interface, writing 77–79
- commands dictionary 136
- complexity, managing 6
- composition vs. inheritance 209
- computer, defined 3
- computers are stupid 3
- condition 33–34
- continue statement 36
- convenience function 56–57
- cProfile 387
- CRUD 100
- csv module, DictWriter 158
- CSV, writing to a file 156
- D**
- data
 - corrupted 55
 - CRUD interface 100
 - CSV format 145
 - exchange formats 146
 - loading and saving 127–131
 - separating from display 94
 - structure 55
- date formatting 158
- debug string 46
- debugging 39–40, 47–51
- decision making 32
- decorator 248–251
- def 53
- design
 - class inheritance vs. composition 208
 - level of abstraction 187
 - object oriented 186–189, 207
 - top down 100
- diamond inheritance 222
- dict dictionary 223
- dictionary 85–86
 - key 85
 - placeholder 87
 - value 85
- diff 97
- dir() method 74
- directory
 - checking existence of 96
 - comparing 89, 95–97
 - root 89
 - searching 79–81
 - test directories 91–93
 - working through with generators 239
- Django 253–286, 358–382
 - adding users 363
 - admin interface 269
 - admin system 267
 - customizing 270
 - auth application 359
 - authentication 358–363
 - Client class 373
 - database 264
 - debugging 381
 - django.views.static.serve 377
 - django-admin 256
 - django-registration 362
 - for loops 274
 - forms 277–280
 - displaying 277
 - submission 278

- Django (*continued*)
 - generic delete 282
 - generic update 282
 - generic views 280
 - get_object_or_404 370
 - if..else 263
 - installation 255
 - logging in 359–362
 - media 377–381
 - servicing 377–379
 - servicing from another server 379–380
 - model forms 280
 - models 265
 - QuerySets 272
 - security issues 280
 - setting up 255–260
 - settings.py 266
 - South application 366
 - template filters 274
 - template syntax cheat sheet 274
 - templates 262
 - TestCase class 373
 - testing 372–377
 - functional testing 373–375
 - response.content 375
 - running tests 375–377
 - unit testing 372–373
 - xUnit tests 373
 - updating database with SQLite 364–367
 - updating interface 371–372
 - URLs 274–277
 - urls.py 259
 - application specific 275
 - using data 271
 - views 261
 - writing web-based apps with 254–260
 - django-admin 256
 - docstring 53
 - doctest 142
- E**
- edge case 165
 - ElementTree 388
 - else command 33
 - Emacs 25
- email
 - creating 160–162
 - sending 162–163
 - email module 159, 163
 - linking to attached images 162
 - MIME structure 159
 - email.parser 164
 - empty list 44
 - encapsulating a program into functions 55–62
 - encapsulation 51
 - enumerate() method 117
 - event handler 295
 - event loop 108
 - event-based programming 295
 - exception 172–179
 - catching 176–178
 - traceback module 178–179
 - tracebacks 173–176
 - exception handler 176
 - Exception object 177
 - extract_error 248
- F**
- Factory 323
 - failure
 - dealing with breakage 166
 - fail loudly 168
 - what can go wrong? 165
 - file
 - closing 83
 - comparing 83–86
 - hash 83–85
 - storing in a dictionary 85–86
 - locating 79–81
 - opening 82–83
 - reading and writing 79–83
 - reading with generators 237–246
 - writing text into 82
 - file() method 82
 - filter function 247
 - Firebug 148
 - Firefox 148
 - first-class function 52
 - for loop 36, 44
 - format_exc() method 178

- function 35–36
 - ** arguments 113
 - advantages of using 52
 - assigning to a variable 109
 - basics 51–53
 - convenience function 56–57
 - docstring 53
 - encapsulation 51
 - first-class function. *See* first-class function
 - hiding program complexity with 51–55
 - input and output 52
 - parameters 53
 - passing and returning 248–251
 - return statement 53
 - shared state 54–55
 - side effects 247
 - variable scope 53–54
- functional programming 246–251
 - decorators 248–251
 - side effects 247
- functional testing in Django 373–375
- functools 250
- G**
- game
 - design 288–318
 - difficulty 314
 - and entertainment 301
 - experimenting with gameplay 318
 - feedback 318
 - must be fun! 318
 - rewarding player 316
 - simulation 308
- generator 233–237
 - calling 235
 - next() method 235
 - reading files 237–246
 - using 237–246
 - working through directories 239
 - yield statement 235
- generator expressions 236
- getattr() method 191
- getmail module 163
- getter 228
- GitHub 387
- Gnome program launcher 20
- gravity 301–308
 - calculating force 305
- group of items, handling 119
- H**
- hash
 - generating 83
 - storing in a dictionary 85–86
- hashlib library 83
- help() method 74
- hexdigest 84
- HTML, id and class attributes 152
- HTTPHandler 151
- HTTPRedirectHandler 151
- Hunt the Wumpus
 - history of 29
 - version 1 36–40
 - version 2 40–68
- I**
- IDLE 25
- if statement 33–34
- image processing 388
- incremental programming 35
- indentation 33
- index 42
- inheritance 206
 - reasons to use 207
 - vs. composition 209
- installation
 - Linux 18–21
 - Mac 21–23
 - Python 18
 - Windows 9–17
- int() function 38
- integrated development environment (IDE), choosing 25
- interface 57–59
- invalid data 165
- is_error 248
- items() method 86
- iter() method 234
- iterator 233–234
 - interface 233
 - next() method 234

J

JSON 388

K

keys() method 86
 known good file 128
 Komodo 26

L

lambda keyword 250
 len() function 38, 82
 library 71–77
 contents of 74–77
 defined 71
 dir() method 74–77
 including everything 73
 including specific functions 73
 including the module 73
 standard library 72
 using 72–74
 list 42
 empty 44
 index 42
 number of lines 82
 of lists 45
 slicing 43
 list comprehension 119–122
 list() method 82
 listiterator 234
 local variable 53
 log file, fields
 getting 242–246
 meaning 241
 log_format 245
 log_regexp 245
 log_split 245
 logging 387
 logging module 220
 logger, creating 221
 loop 34
 for loop. *See* for loop
 while loop. *See* while loop
 lower() method 113

M

mailbox module 163
 maildir 163

manage.py 259
 map function 247
 mbox 163
 method, common to override 232
 mixin class 219–222
 Model, View, Controller 254
 MUD 320, 326–357
 angry list 333–335
 monsters 333–335
 player interaction 335–340
 potential extra features 356
 saving 352–356
 shouting 338
 multiprocessing module 388

N

nesting 33
 new line character 83
 Newton's first law 298
 None 89
 not command 34
 not in command 34
 Notepad 12
 is limited 25
 Notepad++ 25

O

object 54
 object orientation 219–223
 object-oriented design 186–189
 tips 207
 open() method 82
 OpenKomodo 26
 ordering results 93–95
 os library 75
 os.listdir() method 128
 os.path.isdir() method 96
 os.path.join() method 81–82
 os.walk() method 79
 using generators in 237
 overriding methods 232

P

parsing 388
 path 79–81
 finding 81
 separator 81

- pickle module 127
- pickle.dump() method 129
- pickle.load() method 130
- PIL (Python Image Library) 388
- pop 331
- print statement 30
- print_exc() method 178
- profiling code 387
- program
 - architecture 100
 - defined 29
 - encapsulating into functions 55–62
 - overview 29–36
 - simplifying 62
 - testing 91–93
- program plan 86–88
- programming
 - and literacy 1
 - are you done? 164
 - as design 5
 - as exploring 6
 - automation 144
 - BASIC 2
 - defined 2
 - documentation 341
 - functional 246–251
 - decorators 248–251
 - side effects 247
 - glue code 144
 - incremental 35
 - introspection 342
 - is fun 2
 - is made of ideas 5
 - knowing when to stop 98
 - read the code 342
 - simplicity 107
 - specifications 100
 - state machines 347–352
 - using example code 341
 - what can I do with programs? 4
- programming development 383–389
- Project Hosting on Google Code 385
- property 227–229
 - extending 230
- property function 227
- PuTTY 321
- py.test 142
- PyDev 26
- Pyglet
 - .rotation attribute 297
 - aliens, evil 311–316
 - Batch class 311
 - bullets 308–311
 - collision detection 305–308
 - colors 316
 - displaying images 291
 - game design 288–318
 - gravity 301–305
 - installing 289–290
 - key presses 296
 - KeyStateHandler 310
 - loading resources 292
 - scheduler 298
 - sprite rotation 294
 - timers 310
 - update function 297
 - velocity and acceleration 299
 - Window object 290
- pygame module 290
- pygame.resource.image function 292
- Pygletevent-based programming 295
- Pyparsing 388
- Python
 - batteries included 8
 - community 9, 385–386
 - errors
 - file extension, incorrect 24
 - installed in a different place 24
 - syntax error 24
 - finding installation location 24
 - installing
 - Linux 18–21
 - Mac 21–23
 - Windows 9–17
 - is a “real” language 8
 - is easy 7
 - Linux command line 20
 - mailing lists 385
 - recipes 384
 - running
 - from the Linux command line 20
 - from the Linux GUI 18–20

Python, running (*continued*)
 from the Windows command line
 14–17
 on Windows 12–14
 user groups 386
 what it can do? 8

Python Enhancement Proposal (PEP)
 237

Python Image Library (PIL) 388

Python Module of the Week 72

Python Standard Library 384

Python Tutor list 385

PYTHONPATH 255

R

range function 46

raw string 243

raw_input() method 32, 137

read() method 82

readlines() method 82, 152

regression testing 114–127

regular expression 242–246
 quick reference 242

render_to_response function 368

Representational State Transfer (REST)
 274

results
 displaying 94
 ordering 93–95, 118
 storing 94

return statement 53

root 89

round-trip 127

running Python
 error messages 14
 from the Linux command line 20
 from the Linux GUI 18–20
 from the Windows command line 14–17
 Gnome program launcher 20
 on Windows 12–14
 program errors 23
 setting Mac default .py app 22
 updating Mac shell profile 21
 using which python on Linux 24
 Windows path settings 15
 your first program 12

S

saving data, extra cautious 131

saving work 127–131

script, broken, dealing with 166–172

self argument 185

sendmail() method 163

setter 228

shared state 54–55

shell profile, updating 21

shlex 388

slicing lists 43

smtplib module, logging in 162

sorted() method 95

SourceForge 385

SPE 26

special character, creating with backslash
 81

specification 100

Sprite 292

SQLite 365

standard library 70, 72

state machine 347–352

state, storing between function calls 233

StopIteration exception 233

string
 .isdigit() method 136
 ljust (left justification) 117
 raw 243

subprocess module 388

super() method 221–223

syntax error 24

sys module 78–79

sys.argv 78

sys.exit 78

T

Telnet 321

tempfile module 170

temporary variable 46

testing 91–93
 for bad input 134
 functional testing in Django 373–375
 functional tests 102
 limitations of unit testing 123
 manual tests 101
 regression testing 114–127, 134

testing (*continued*)
 response.content 375
 running tests in Django 375–377
 stress testing 169
 Test Driven Development 102
 unit testing in Django 372–373
 unit tests 99, 102
 unittest 142
 user input 109–110
 user interfaces 107–109
 xUnit style 373

text editor, choosing 25
 text input, limitations of 137
 textwrap module 125
 todos variable 104
 traceback 173
 traceback module 178–179
 troubleshooting 23–24
 try..except block 176
 tuple 90
 Twisted 320–357
 API documentation 341
 AuthenticatingTelnetProtocol 347
 authentication 342–347
 class structure 343
 connectionLost 323
 connectionMade 323
 Deferreds 348
 documentation 341
 event loop 328
 exploring the code 340–342
 Factory 323
 installing 320
 lineReceived 323
 protocol 321
 scheduling tasks 328
 StatefulTelnetProtocol 322
 storing passwords 346
 task.LoopingCall 328
 Twistedtask.LoopingCall 328
 type, emulating 229–232

U

unit test 103–107
 independence 104

unit testing 102
 in Django 372–373
 limitations of 123
 unittest 142
 update() method 84
 urllib2
 handling HTTP redirects 151
 in python 3.0 152
 urllib2 module 150
 urls.py 259
 user groups 386
 user input
 asking for 32
 testing 109–110
 user interaction 59–60
 user interface, testing 107–109
 user_passes_test function 248

V

values() method 86
 variable 31–32
 assigning function to 109
 local variable. *See* local variable
 temporary 46
 types 32
 variable scope 53–54
 vectors and angles 298–301
 Vim 25

W

web framework 253
 web page, downloading 150
 while loop 34
 Wing IDE 26
 wrapping x and y 300
 write() method 82
 writelines() method 82

X

xmllib 388
 xUnit 373

Z

zip function 246