

INDEX

Symbols

- + (addition), 30
- * (asterisk) operator, 157
- .bmp* (Bitmap) image files, 244
- + (concatenation operator), 101–102
- / (division), 30
- == (equality operator), 77
- > (greater than), 79
- >= (greater than or equal to), 79
- # (hash mark), for comments, 33
- != (inequality operator), 78
- < (less than), 79
- <= (less than or equal to), 79
- % (modulo operator), 120–121
- * (multiplication), 30
- \n (newline), 26
- ! (not), 78
- += operator, 119
- .py* file extension, 20
- [] (square brackets), 38
- (subtraction), 30
- \t (tab), 26

A

- addition (+), 30
- aliases, 156, 157
- alice.py*, 203–205
- Alien Invasion project. *See also*
 - Pygame
 - aliens
 - Alien class, 267
 - building fleet of, 269–275
 - changing directions, 278
 - checking edges, 278
 - creating, 266
 - creating rows of, 270

- drawing to the screen, 268
- dropping, 278
- fitting in a row, 269
- movement, 276–279
- multiple rows of, 273
- reaching bottom of screen, 288
- rebuilding fleet of, 282
- background color, setting, 242
- bullets
 - Bullet class, 258–259
 - checking with print statements, 262
 - deleting old, 261
 - firing, 260
 - limiting the number of, 262
 - making larger, 281
 - settings, 257
 - speeding up, 283
- classes
 - Alien, 267
 - Bullet, 258–259
 - Button, 292
 - Scoreboard, 301
 - Settings, 243
 - Ship, 245
- collisions, 280
 - alien-bullet, 280
 - alien-ship, 284–288
- ending a game, 288–289
- keyboard shortcuts, 266
- levels
 - adding, 299–301
 - modifying speed settings, 299
 - resetting speeds, 300
- planning, 236

Alien Invasion project, *continued*

Play button

- adding, 292–298
- Button class, 292
- deactivating, 297
- drawing, 294
- hiding the mouse
 - cursor, 298
- resetting the game, 296
- starting the game, 295

reviewing, 266

scoring

- displaying the level, 310–312
- displaying the number of ships, 313–316
- displaying the score, 301
- increasing point values, 306
- making a scoring system, 301–316
- rounding and formatting the score, 307
- Scoreboard class, 301
- scoring all hits, 305
- tracking the high score, 308
- tracking the score, 301

ship

- adjusting speed, 253–254
- continuous movement, 250–253
- finding an image, 244
- limiting range, 255
- Ship class, 245

alien.py, 96–100

aliens.py, 109–111

americas.py, 367

amusement_park.py, 84–87

and keyword, 79–80

API (application programming interface), 377

call, 378

- processing response, 379–383
- requesting data using, 378
- summarizing results of, 382

for Hacker News, 390–393

rate limits, 383

apostrophe.py, 28

append() method, 41–42

application programming interface.
See API (application programming interface)

arguments, 135–141. *See also* functions

arithmetic, 30

as keyword, 156, 157

assert methods, 218, 222–223

asterisk (*) operator, 157

attributes, 163

accessing, 164

default values, 168

modifying values, 168–171

B

banned_users.py, 81

Bash, 475

bicycles.py, 38–40

birthday.py, 31

bitmap (.*bmp*) image files, 244

body

of a function, 134

of an HTML file, 458

Boolean values, 81

Bootstrap, 457

built-in functions, 490

C

calling

functions, 133, 134, 139–140

methods, 165

CamelCaps, 186

car.py, 167–171

car.py module, 179–181, 183–184

cars.py, 47–49, 76

child classes, 172. *See also* classes:
inheritance

cities.py, 125

classes, 161–162

attributes, 163

accessing, 164

default values, 168

modifying values, 168–171

creating, 162–164

- importing
 - all classes from a module, 182–183
 - entire module, 182
 - module into a module, 183–184
 - multiple classes, 181–182
 - single class, 179–180
- inheritance, 172–178
 - attributes and methods, 174–175
 - `__init__()` method, 172–173
 - instances as attributes, 175–177
 - overriding methods, 175
 - in Python 2.7, 173
 - `super()` function, 173
- methods
 - calling, 165
 - `__init__()`, 163
- modeling real-world objects, 177–178
- multiple instances, creating, 165
- styling guidelines, 186–187
- colormap_colors.py*, 330
- colors
 - hex format, 373
 - Pygal themes, 374
- comma-separated value files. *See* CSV (comma-separated value) files
- comment.py*, 33
- comments, 33–34
- comparison operators, 78–80
- concatenation operator (+), 101–102
- conditional tests, 76–81
- confirmed_users.py*, 128
- counting.py*, 122, 126
- countries.py*, 365
- country codes, 365–367
- country_codes.py*, 365
- CSV (comma-separated value) files, 350–356
 - error-checking, 359
 - parsing headers, 350
 - reading data, 352

D

- databases. *See* Django: databases; Heroku: databases
- data mining, 321
- data visualization, 321–322. *See also* matplotlib; Pygal
- datetime module, 354–355
- Death Valley, weather data, 350, 359–362
- debugging tips, 499–500
- decorators, 447
- default values
 - class attributes, 168
 - function parameters, 138
- def keyword, 134
- del statement, 100
- dice_visual.py*, 343–346
- dictionaries
 - defining, 96
 - empty, 98
 - formatting larger, 101
 - key-value pairs
 - adding, 97
 - removing, 100
 - looping through, 102
 - keys, 104–106
 - keys in order, 106–107
 - key-value pairs, 103–104
 - values, 107–108
- values
 - accessing, 97
 - modifying, 99–100
- Die class, 340
- die.py*, 340
- die_visual.py*, 341–343
- dimensions.py*, 70–71
- div (HTML), 460
- division (/), 30
- division.py*, 200–203
- Django, 397. *See also* Heroku; Learning Log project
 - admin site, 406–408
 - Bootstrap, 457
 - current working directory, 469

- Django, *continued*
 - data
 - associating with a user, 453
 - connecting to users, 448
 - restricting access to, 446–453
 - databases
 - creating, 401
 - foreign keys, 408
 - many-to-one relationships, 408
 - migrating, 401, 409, 450–451
 - queries, 423
 - querysets, 410
 - forms, 428–438
 - cross-site request forgery, 431
 - displaying, 431
 - GET and POST requests, 430
 - ModelForm, 428, 432
 - processing, 430, 434
 - validation, 428
 - widgets, 432
 - hashes (for passwords), 406
 - HTTP 404 error, 452
 - INSTALLED_APPS, modifying, 405
 - jQuery, 457
 - localhost, 402
 - @login_required, 447
 - login view, 440
 - manage.py*, 400
 - models, 403–405
 - privileges, 406
 - runserver command, 401
 - static files, 467
 - superuser, setting up, 406
 - templates
 - anchor tags, 417
 - block tags, 417
 - context, 420
 - filters, 423
 - linebreaks filter, 423
 - template tags, 417
 - third party apps, 456
 - URLs
 - namespaces, 417
 - patterns, 414
 - regular expressions, 414, 422
 - user ID values, 449
 - views, 412
 - web server gateway interface, 401
 - django-bootstrap3 app, 456
 - docstrings, 134
 - dog.py*, 162–166
 - dot notation, 155, 164
- E**
- Einstein, Albert, 29
 - electric_car.py*, 172–178
 - electric_car.py* module, 183
 - elif statement, 84–88
 - else statement, 83–84
 - Emacs, 497
 - epoch time, 384
 - equality operator (==), 77
 - even_numbers.py*, 62
 - even_or_odd.py*, 121
 - event loops, 241–242
 - exceptions, handling, 200
 - deciding which errors to report, 207
 - else blocks, 202–203
 - failing silently, 206–207
 - FileNotFoundError, 203–204
 - to prevent crashes, 201–202
 - try-except blocks, 200–201
 - ZeroDivisionError, 200
- F**
- favorite_languages.py*, 100–102, 104–108, 112
 - FileNotFoundError, 203–204
 - file_reader.py*, 190–194
 - files
 - closing, 191
 - large, 195–196
 - opening, 190
 - paths, 191–192
 - reading
 - entire files, 190–191
 - line by line, 193
 - by making a list of lines, 194

- read/write modes, 198
- working with contents, 194
- writing to
 - by appending, 199
 - empty files, 197
 - multiple lines, 198
- flags, 124
- floats, 30–31
- foods.py*, 67–69
- for loops, 53–60
- functions, 133
 - arguments, 135–141
 - arbitrary, 151–152
 - avoiding errors with, 140–141
 - keyword, 137–138
 - lists as, 147–150
 - optional, 142–144
 - order of, 137
 - positional arguments, 136–137, 152
 - built-in, 490
 - calling, 133, 134, 139–140
 - multiple times, 136–137
 - defining, 134
 - dictionaries, returning, 144–145
 - import statements, 154
 - lists in
 - modifying, 147–149
 - preventing modifying, 149–150
 - modules, storing in, 154–157. *See also* modules
 - parameters, default values
 - for, 138
 - passing information to, 134
 - return values, 141–146
 - styling, 158

G

- games. *See* Alien Invasion project; Pygame
- Geany, 492–494
 - commenting out blocks, 494
 - customizing settings, 493
 - indenting and unindenting blocks, 494

- installing
 - on Linux, 492
 - on Windows, 492–493
 - running Python programs, 493
- get-pip.py*, 238. *See also* pip
- GET requests, 430
- Git, 378, 505–506
 - branches, 508
 - commits, 378, 508
 - checking out previous, 512
 - making, 509
 - detached HEAD, 512
 - files
 - adding, 508
 - ignoring, 472, 507
 - HEAD, 512
 - installing, 506
 - log, checking, 509
 - projects
 - checking status of, 508
 - making, 507
 - repositories
 - deleting, 513
 - initializing, 507
 - reverting changes, 511–512
- GitHub, 378. *See also* Git
- greater than (>), 79
- greater than or equal to (>=), 79
- greeter.py*, 118–119, 134–135
- greet_users.py*, 147
- gunicorn, 469, 470–471

H

- Hacker News, 390
- hash mark (#), for comments, 33
- head, of an HTML file, 458
- HEAD (Git), 512
- Hello World, 4
- hello_world.py*, 19–23
- Heroku. *See also* Django; Learning Log project
- Bash, 475
- databases
 - migrating, 474
 - setting up, 474
 - using Postgres, 469

- Heroku, *continued*
 - making an account, 466
 - Procfile*, 469
 - projects
 - deleting, 481
 - pushing to, 473
 - securing, 476
 - Python runtime, specifying, 468
 - settings.py*, modifying for, 468
 - superuser, creating, 475
 - Toolbelt, installing, 466
 - user-friendly URLs, 476
 - wsgi.py*, modifying for, 470
 - hex format, for colors, 373
 - highs_lows.py*, 350–361
 - hn_submissions.py*, 391–393
 - Homebrew, 487
- I**
- IDLE, 496–497
 - commenting out code, 497
 - customizing settings, 497
 - installing
 - on Linux, 496
 - on OS X, 496
 - on Windows, 497
 - if statements
 - and keyword, 79–80
 - Boolean expressions, 81
 - checking for
 - empty lists, 91
 - equality (`==`), 76–78
 - inequality (`!=`), 78
 - items in a list, 80–81
 - items not in a list, 81
 - special items, 90–91
 - elif statement, 84–88
 - else statement, 83–84
 - if-elif-else chains, 84–93
 - lists and, 89–92
 - numerical comparisons, 78–80
 - or keyword, 80
 - simple, 82–83
 - styling guidelines, 94
 - testing multiple conditions,
 - 87–88
 - import *, 157
 - import this, 34
 - indentation errors, 57–60
 - index errors, 51
 - inequality operator (`!=`), 78
 - infinite loops, 126–127
 - inheritance, 172. *See also* classes:
 - inheritance
 - input() function
 - numerical input, 119–120
 - prompts, 118–119
 - insert() method, 42
 - int, 31
 - IRC (Internet relay chat), 502–503
 - itemgetter() function, 391–392
 - items() method, 103
- J**
- jQuery, 457
 - json.dump() function, 209
 - JSON files
 - converting strings to numerical values, 364
 - extracting data, 363
 - json.load() function, 210
 - jumbotron, 461
- K**
- keys() method, 104
 - key-value pairs, 96. *See also* dictionaries
 - keyword arguments, 136, 137–138
 - keywords, 489
- L**
- language_survey.py*, 224
 - Learning Log project, 397. *See also* Django; Heroku
 - deployment
 - commits, using Git, 471–473
 - custom error pages, 478–480
 - get_object_or_404()
 - method, 480
 - ignoring files in Git, 472
 - making a static file
 - directory, 470
 - ongoing development, 480
 - Procfile*, 469

- requirements.txt*, 467
 - SECRET_KEY setting, 481
 - using gunicorn locally, 470
- HTML headers, defining, 458
- logging out, 442–443
- login page, 440–442
- migrating the database,
 - 450–451
- styling, 456
 - Bootstrap, 457
 - collapsible navigation, 459
 - jumbotron, 461
 - navigation bar, 458–460
 - selectors, 459
- registration page, 443–446
- users app
 - default login view, 440
 - displaying messages to
 - logged-in users, 441
 - logging in a user, 445
 - UserCreationForm, 445
- len() function, 49
- less than (<), 79
- less than or equal to (<=), 79
- Linux
 - Geany (text editor), 6, 492–494
 - Python
 - checking installed version,
 - 5, 486
 - installing, 485–486
 - setting up, 5–8
 - running Hello World, 6–7
 - terminal
 - running commands from, 8
 - running programs from, 16
 - troubleshooting installation
 - issues, 15
- lists, 37–38
 - append() method, 41
 - as arguments, 147–150
 - copying, 67–69
 - del, 43
 - elements
 - accessing, 38–39
 - adding, 41–42
 - modifying, 40
 - empty, 42
 - for loops, 53–60
 - if statements, 89–92
 - indentation errors, 57–60
 - indexes, 39
 - errors, 50–51
 - negative, 39
 - insert() method, 42
 - len() function, 49
 - list comprehensions, 63–64
 - max() function, 63
 - min() function, 63
 - naming, 38
 - numerical, 61–64
 - pop() method, 43–45
 - range() function, 62–63
 - remove() method, 45
 - removing all occurrences of a
 - value, 129
 - reverse() method, 49
 - slices, 65–67
 - sorted() function, 48
 - sort() method, 47
 - sum() function, 63
- localhost, 402
- logical errors, 58
- lower() method, 24
- lstrip() method, 27

M

- magicians.py*, 54–57
- magic_number.py*, 79
- making_pizzas.py*, 155–157
- matplotlib
 - formatting plots
 - axes, 337
 - color, 329–330
 - labels, 324–325
 - line thickness, 324–325
 - shading, 358
 - size, 338
 - installing, 322–323
 - plot() function, 324
 - plotting
 - dates, 355
 - multiple data series, 357
 - pyplot module, 324
 - saving plots, 331
 - scatter plots, 326–329
 - simple line graphs, 324

methods, 24, 165, 174–175. *See also*
 classes
ModelForm, 428, 432
modules, 154–157
 aliases for, 157
 functions
 aliases for, 156
 importing all, 157
 importing specific, 156
 importing an entire
 module, 154
modulo operator (%), 120–121
motorcycles.py, 41–46
mountain_poll.py, 130
mpl_squares.py, 324–326
multiplication (*), 30
my_car.py, 180
my_cars.py, 181–184
my_electric_car.py, 181

N

name errors, 21–23
name_function.py, 216–221
name.py, 24–25
names.py, 216
na_populations.py, 368
nesting
 dictionaries in dictionaries,
 113–114
 dictionaries in lists, 109–111
 lists in dictionaries, 111–113
newline (\n), 26
not (!), 78
number_reader.py, 209
numbers
 arithmetic, 30
 avoiding type errors, 31–32
 comparisons, 78–80
 exponents, 30
 floats, 30–31
 integers, 30–32
 order of operations, 30
 str() function, 32
numbers.py, 61
number_writer.py, 209

O

object-oriented programming, 161.
 See also classes
open() function, 190
operator module, 391–392
or keyword, 80
OS X
 Python
 checking installed version,
 8, 486–487
 installing, 486–488
 setting up, 8–10
 running Hello World, 10
 Sublime Text (text editor), 9,
 494–496
 terminal
 running commands from, 9
 running programs from, 16
 troubleshooting installation
 issues, 15

P

parameters, 135, 138–139
parent classes, 172. *See also* classes:
 inheritance
parrot.py, 118, 122–125
pass statement, 206
PEP 8, 72–73
person.py, 144–146
Peters, Tim, 34
pets.py, 129, 136–140
pi, 190–196
pip, 236
 checking for, 237
 get-pip.py, 238
 installing, 238
pi_string.py, 194–196
pizza.py, 111, 151–152
planning a project, 236
players.py, 65–67
pop() method, 43–45
positional arguments, 135,
 136–137, 152
Postgres, 469
POST requests, 430

- printing_models.py*, 147–150
- print statements, 24
 - long, 102
 - in Python 2, 29
- privileges, 406
- Profile*, 469
- Project Gutenberg, 208
- prompts, for user input, 118–119
- .py* file extension, 20
- Pygal, 322
 - charts
 - histograms, 342–343
 - links, adding, 390
 - styling, 386–388
 - tooltips, adding, 387
 - color themes, 374
 - installing, 340
 - plotting dictionaries, 388
 - rolling dice, 339
 - WorldMap, 367–374
 - grouping countries, 371–372
 - plotting numerical data, 368–372
 - styling, 372–374
- Pygame. *See also* Alien Invasion
 - project
 - bitmap (*.bmp*) image files, 244
 - colors, 242–243
 - creating an empty window, 241
 - displaying text, 293
 - drawing images to the screen, 246
 - groups
 - emptying, 282
 - storing elements in, 259
 - updating all elements in, 260
 - installing, 236–240
 - positioning images with
 - rect, 245
 - responding to input, 241–242
 - keypresses, 249–250
 - mouse clicks, 295–296
 - screen coordinates, 246
 - surfaces, 241
- pypplot module, 324
- Python
 - built-in functions, 490
 - documentation, 501
 - installing
 - on Linux, 485–486
 - on OS X, 487
 - on Windows, 488–489
 - interpreter, 20
 - keywords, 489
 - PEP 8, 72–73
 - standard library, 184–185
 - terminal session, 4
 - version, 486
 - Zen of, 34–36
- Python 2.7
 - creating classes in, 164
 - division of integers in, 32
 - print statements in, 29
 - raw_input() function, 121
- Python Enhancement Proposal (PEP), 72
- python_repos.py*, 379–390

Q

- queries
 - in databases, 423
 - on GitHub (q=), 378–379
- queriesets, in databases, 410
- quit values, 122–124

R

- random_walk.py*, 332
- random walks, 331–332
 - coloring points, 335–336
 - fill_walk() method, 332
 - multiple walks, generating, 334–335
 - plotting, 333–334
 - RandomWalk class, 332
 - starting and ending points, plotting, 336
- range() function 61–63
- readlines() method, 194
- read() method, 191
- rect, positioning images with, 245
- reddit, 502
- refactoring, 212–214
- remember_me.py*, 210–214
- requests package, 379

- resources
 - debugging tips, 499–500
 - documentation, 501–502
 - IRC (Internet relay chat), 502–503
 - channels, 503
 - culture, 503
 - making an account, 502
 - reddit, 502
 - Stack Overflow, 501
- return values, 141
- rollercoaster.py*, 120
- rolling dice
 - analyzing results, 341
 - different sizes, 345
 - two dice, 343
- rubber duck debugging, 500
- rw_visual.py*, 333–338

S

- scatter_squares.py*, 327–330
- SECRET_KEY setting, 481
- setUp() method, 227
- Sitka, Alaska, weather data, 350–362
- slice, 65
- sorted() function, 48
- sort() method, 47
- split() method, 204
- SQLite, 401
- square brackets ([]), 38
- squares.py*, 62, 64
- Stack Overflow, 501
- storing data
 - json.dump() method, 209
 - json.load() method, 209
 - saving and reading data, 210–211
- str() function, 32
- strings
 - changing case, 24
 - concatenating, 25
 - newlines in, 26
 - single and double quotes, 23
 - tabs in, 26
 - whitespace in, 26–27
- strip() method, 27

- style guidelines, 72
 - blank lines, 73
 - CamelCaps, 186
 - classes, 186
 - functions, 158
 - if statements, 94
 - indentation, 72
 - line length, 73
 - PEP 8, 72
- subclasses, 173. *See also* classes: inheritance
- Sublime Text, 494
 - commenting out code, 496
 - configuring, 495
 - indenting and unindenting code blocks, 496
 - installing
 - on Linux, 494
 - on OS X, 494
 - on Windows, 495
 - running Python programs, 495
- subtraction (-), 30
- superclasses, 173. *See also* classes: inheritance
- superuser
 - in Django, 406
 - in Heroku, 475
- survey.py*, 223
- syntax errors, 28
- syntax highlighting, 20

T

- tab (\t), 26
- testing code
 - adding tests, 221
 - assert methods, 218, 222–223
 - failing tests, 218–220
 - full coverage, 217
 - functions, 216–221
 - passing tests, 217–218
 - test case, 217
 - testing classes, 222–228
 - examples, 223–226
 - setUp() method, 227
 - unittest module, 215, 217
 - unit tests, 217
- test_name_function.py*, 217–221

- test_survey.py*, 225–228
- text editors, 491–497
- title() method, 24
- Toolbelt, Heroku, 466
- toppings.py*, 78, 87–92
- traceback, 22
- tracking project files, 378, 471–473.
 - See also Git
- try-except blocks, 200–201. See also exceptions, handling
- tuples
 - defining, 69
 - for loop, 70
 - writing over, 71
- type errors, 31–32, 70

U

- unittest module, 215, 217
- unit tests, 217
- Unix time, 384
- upper() method, 24
- user_profile.py*, 152
- user.py*, 103–104
- users.py*, 113

V

- values
 - key-value pairs, 96. See also dictionaries
 - of variables, 20
- values() method, 107
- variables
 - defining, 20
 - name errors, 21–23
 - naming conventions, 21
- version control system, 378.
 - See also Git
- vim, 497
- virtual environment
 - deactivating, 399
 - installing virtualenv, 399
- voting.py*, 83–84

W

- weather data, 350–362
- web framework, 397
- while loops
 - active flag, 124–125
 - break statement, 125
 - infinite, 126–127
 - moving items between lists, 128
 - quit values, 122–124
- whitespace
 - adding, 26
 - stripping, 26–27
- Windows
 - Geany (text editor), 12–14, 492–494
 - Python
 - installing, 11, 488–489
 - setting up, 10–14
 - running Hello World, 14
 - terminal
 - running commands from, 11–12
 - running programs from, 16–17
 - troubleshooting installation issues, 15
- with keyword, 190
- word_count.py*, 205–207
- world_population.py*, 363–374
- write_message.py*, 197–199
- write() method, 198

Z

- Zen of Python, 34–36
- ZeroDivisionError, 200