

INDEX

SYMBOLS & NUMBERS

- + (addition operator), 48–49
- += (addition shorthand), 62
- * (asterisk), importing all functions with, 241
- / (division operator), 58
- /= (division shorthand), 62
- "" (double quotation marks), for strings, 66
- = (equal sign), assigning values to variables with, 28
- == (equal to), 84
- ** (exponential operator), 60–61, 91–92
- > (greater than), 88
- >= (greater than or equal to), 89–90
- # (hash mark), for comments, 35
- < (less than), 88–89
- <= (less than or equal to), 89–90
- * (multiplication operator), 58
- *= (multiplication shorthand), 62
- != (not equal to), 86–87
- " (single quotation marks), for strings, 66
- [] (square brackets), for defining lists, 168
- (subtraction operator), 48
- = (subtraction shorthand), 62
- """ (triple quotation marks), for docstrings, 152–153
- 2D lists, 208–213, 216
- 3D lists, 218–225

A

- a (append permission), 233
- addition operator (+), 48–49
 - shorthand (+=), 62
- aliases, for modules, 241
- and operator, 93

- API (application programming interface), Minecraft Python
 - installing on Mac, 15
 - installing on Windows, 6
- append permission (a), 233
- append() function, 171–172
- application programming interface. *See* API
- arguments, 34, 147–148
 - line breaks in, 153
 - math operators in, 54–55
- arrays. *See* lists
- asterisk (*), importing all functions with, 241
- attributes, 257. *See also* variables
 - accessing, 259–260
 - class, 271–273

B

- block hits program, 180–182, 196–198
 - scoreboard, 192–194, 205–206
- blocks
 - changing, 52–53, 138–139, 196–198
 - finding highest, 90
 - identifying, 85
- IDs
 - cheat sheet, 283
 - finding by, 97–98, 186–187, 207–208
 - reminder program, 155–156
- moving, 163–165
- placing, 49, 55–56
 - by user input, 74–75
- random, 160–161, 183
- replacing, 173–174
- stacking, 49–50

blocks, *continued*
state, 158–159
wool, setting color by name,
158–159
Boolean operators. *See* logical
operators
Boolean values, 82–83
break statements, 139, 207
building quickly, 55–57

C

chat
persistent, 139–140
posting to, 67–68, 69–70
usernames, 72–73
cheat sheet, block IDs, 283
choice() function, 182
class attributes, 271–273. *See*
also attributes, global
variables
classes, 257–260, 273–274
close() function, 233–234, 247–248
color of wool blocks, setting by
name, 158–159
command prompt, 21, 23–24
comments, 35, 152–153
comparators, 83–91, 104–105,
131–132
concatenation, 71–72, 83
conditions, 81, 104–105, 131–132
connecting to Minecraft, 34
constructor, 259
coordinates, 31–32
copying structures, 225–229,
242–246, 248–252
count variables, 124, 127–128
crater program, 105–106
curse program, 128–129

D

dance floor, generating, 135–137
data. *See also* files
storing with variables, 28
types, 31
debugging, 42–44
decimal values, 37–38
decrementing values, 128

def keyword, 146
del keyword, 172–173
delays, setting in programs, 39–40
dictionaries. *See also* *shelve* module
defining, 188–189
items
accessing, 189
adding, 191–192
changing, 191–192
deleting, 192
looping over, 205
readability, 205
diving contest program, 132–134
division operator (/), 58
shorthand (/=), 62
docstrings, 152–153
double quotation marks (""),
for strings, 66
dump() function, 239–240

E

elif statements, 109–110, 112–113
else statements, 107, 141, 206–207
else-if statements. *See* *elif*
statements
equal to (==), 84
equal sign (=), assigning values to
variables with, 28
errors
debugging, 42–44
handling, 76–78
index, 168–169
scope, 162
syntax, 30
type, 147–148, 154–155
exception handling, 76–78
exponential operator (**), 60–61,
91–92
expressions, 47–48

F

False (Boolean value), 82
files, 231–235
opening, 232–233, 247
reading, 234–235
saving, 233–234, 247–248

- shelve module, using with, 247–248
- writing to, 233–234
- Flask module, 253–255. *See also* modules, pip
- floats, 37–38
 - converting to strings, 71–72
- flower trail, creating, 130–131
- forest, building, 148–150
- for loops, 195–196
 - with dictionaries, 205
 - generating 2D lists with, 216
 - with multidimensional lists, 208–213, 218–225
- for-else loops, 206–207
- functions, 145. *See also* methods
 - arguments, 147–148
 - calling, 146–147
 - defining, 146
 - returning values with, 153–155, 179, 266

G

- getBlock() function, 85
- getHeight() function, 90
- getPos() function, 56
- getTilePos() function, 51
- ghost structures
 - castle, 266–268
 - hotel, 275–277
 - house, 263–265
 - tree, 280–281
 - village, 269–271
- gifts program, 110–111
- global variables, 162–163
- greater than (>), 88
- greater than or equal to (>=), 89–90

H

- hardcoded values, 68
- hash marks (#), for comments, 35
- "Hello, Minecraft World", posting to chat, 67–68
- hot and cold game, 141–143

I

- IDLE, 20–24
- if statements, 103–105
 - with Boolean operators, 119–120
 - in functions, 157–158
 - with lists, 185–186
 - nested, 115, 137
 - with range checks, 117
- importing modules, 39–40, 238–241
- immutable
 - strings, 175
 - world, 82–83, 108–109
- in operator, 185–186
- increment, 127–128
- indentation, 76, 104, 146
- index, of a list, 168–169, 213–214, 223–225
- infinite loops, 127–128
- inheritance, 273–275, 278–280
- __init__() method, 258–260, 278–280
- input
 - numbers only, 77–78
 - placing blocks by, 74–75
- input() function, 68–69
- installation. *See* Mac, Raspberry Pi, Windows
- int() function, 74
- integers, 31
 - converting to a string, 71–72
 - range checks, 117, 135
- iteration, 123–124

J

- Java
 - installing on Mac, 14
 - installing on Windows, 4–5
- joining strings, 71–72

K

- keys, in dictionaries, 188–189

L

- lava trap, setting, 52–53
- len() function, 179

- less than (<), 88–89
- less than or equal to (<=), 89–90
- lists, 167–169, 208–213
 - copying, 183–185
 - creating, 168
 - generating with `range()`, 198–199, 200–201
 - index positions of, 168–169, 213–214, 223–225
 - items in
 - accessing, 168–169
 - adding, 171–172
 - changing, 169
 - deleting, 172–173
 - finding, 185–186
 - inserting, 172
 - length, 179
 - slicing, 184–185
 - three-dimensional, 218–225
 - two-dimensional, 208–213, 216
- list slice, 184–185
- `list()` function, 200–201
- `load()` function, 241
- local variables, 162–163
- logical operators, 92–100
 - and, 93
 - and if statements, 119–120
 - not, 96–97
 - or, 95
 - order of operations, 98–99
 - and while loops, 134–135
- loops. *See* for loops, while loops

M

- Mac, setup instructions, 11–18
- magic wand program, 196–198
- math module, 142
- math operators, 48–58
 - addition (+), 48–49
 - exponential (**), 60–61, 91–92
 - division (/), 58
 - multiplication (*), 58
 - order of operations, 61
 - shorthand, 62
 - subtraction (-), 48
- methods, 257, 261–263. *See also*
 - classes, functions
 - adding to subclasses, 275

- inheritance, 274
 - overriding, 278–280
 - returning values with, 266
- Midas touch program, 138–139
- Minecraft
 - API (application programming interface)
 - installing on Mac, 15–16
 - installing on Windows, 6–7
 - connecting programs to, 34
 - game
 - installing on Mac, 12–13
 - installing on Windows, 2–3
 - playing offline
 - on Mac, 18
 - on Windows, 9–10
 - server
 - installing on Mac, 15–16
 - installing on Windows, 6–7
 - worlds, creating new
 - on Mac, 17
 - on Windows, 8–9
- modules, 238–241
 - installing with pip, 252–253
 - nicknames for, 241
 - pickle, 238–241
 - shelve, 247–248
 - time, 39–40
- moving block program, 163–165
- multiplication operator (*), 58
 - shorthand (*=), 62

N

- nicknames, for modules, 241
- night vision sword program, 186–187
- not equal to (!=), 86–87
- not operator, 96–97

O

- object-oriented programming, 257–258
- objects, 257–260, 269–270
- offline, playing Minecraft
 - on Mac, 18
 - on Windows, 9–10
- `open()` function, 232–233, 247

operators. *See* logical operators,
math operators
or operator, 95
order of operations
logical operators, 98–99
math operators, 61
OS X, setup instructions, 11–18

P

package manager, 252
parameters, of functions, 148
permissions, for files, 232–233, 239
pickle module, 238–241
pillars, building, 202–203
pip, installing modules with,
252–253
pixel art, 214–215
pollBlockHits() function, 180–182,
196–198
position, of player, 31–33. *See also*
teleporting
changing, 34
finding, 51, 56
in specific environments,
85–86, 87–88, 90–91,
93–96
in specific locations, 91–92,
100–101
highest and lowest, 169–171
postToChat() function, 67–68
print() function, 66–67
progress bar, 173–174
pyramid, building, 203–204
Python
installing on Mac, 13
installing on Windows, 3–4
Python shell, 20–21, 23–24

Q

quotation marks
for docstrings, 152–153
for strings, 66

R

r (read permission), 233
r+ (read-and-write permission), 233

randint() function, 62–63
random module, 62–63, 182–183
range checks, 117, 135
range() function, 198–199, 200–201
Raspberry Pi, setup instructions,
18–19
read-and-write permission (r+), 233
read permission (r), 233
read() function, 234
readline() function, 234–235
refactoring, 150–152
return keyword, 153–155, 179, 266
reversed() function, 201–202
running a program, 36

S

scope, of variables, 162–163
scoreboard, for block hits game,
192–194, 205–206
secret passage, building, 115–116
server
installing on Mac, 15
installing on Windows, 6
setBlock() function, 49, 158–159
setBlocks() function, 55–56
setPos() function, 38
setTilePos() function, 34–35
setting() function, 82–83
setup instructions
for Mac, 11–18
for Raspberry Pi, 18–19
for Windows, 2–11
shell, 20–21, 23–24
shelve module, 247–248
shorthand operators, 62
shower program, 120–122
sightseeing guide, creating,
190–191
single quotation marks ('),
for strings, 66
sleep() function, 39–40
slices, of lists, 184–185
sliding program, 177–178
smashing, preventing, 82–83,
108–109
Spigot
on Mac, 15–18
on Windows, 6–11

- spires, creating, 58–60
- sprint record, 78–80
- sqrt() function, 142
- square brackets ([]), for defining lists, 168
- square root, calculating, 142
- stairs, building, 199–200
- state, of blocks, 158–159
- statements, 29–30, 47–48
- str() function, 71–72, 83
- strings, 66
 - accessing characters in, 175
 - concatenating, 71–72
 - converting to integers, 74
- subclasses, 273–275, 278–280
- subtraction operator (-), 52
 - shorthand (-=), 62
- super jump program, 63–64
- superclasses, 273–275, 278–280
- survival mode
 - on Mac, 18
 - on Windows, 10–11
- sword
 - hits, 180–182, 196–198
 - magic wand, 196–198
 - night vision, 186–187
- syntax, 29–30

T

- teleporting, 31–35, 40–42
 - by location name, 190–191, 260–261
 - by point score, 113–114
 - precisely, 38–39
 - to random locations, 125–126
 - restrictions, 118–119
- text. *See* files, strings
- text editor, 21–23
- three-dimensional lists, 218–225
- throwing an exception, 76
- time module, 39–40
- to-do list, 235–237
- triple quotation marks ("""), for docstrings, 152–153
- True (Boolean value), 82
- try-except statements, 76–78
- tuples, 175–176, 179

- two-dimensional lists, 208–213, 216
- TypeError, 147–148, 154–155

U

- UnboundLocalError, 162
- usernames, adding to chat, 72–73

V

- values
 - in dictionaries, 188–189, 191–192
 - of variables, 28
- variables, 28–31, 168
 - assigning values to, 28
 - changing values of, 31
 - global, 162–163
 - local, 162–163
 - naming, 28–29
 - syntax, 29–30

W

- w (write permission), 232–233
- waiting, in programs, 39–40
- wand, magic, 196–198
- watery curse program, 128–129
- weather-worn wall, building, 217–218
- website, creating with Flask, 253–255
- while loops, 123–124
 - conditions, 131–132
 - ending, 127–128, 139
 - with if statements, 137
 - infinite, 127–128, 130
 - with return statements, 160
- while-else statements, 141
- Windows, setup instructions, 2–11
- wool blocks, setting color by name, 158–159
- worlds (Minecraft), creating new
 - on Mac, 17
 - on Windows, 8–9
- write permission (w), 232–233
- write() function, 233–234

X

- x, y, and z coordinates, 31–32