Learn to Program with Minecraft

## **INDEX**

### **Symbols & Numbers**

+ (addition operator), [52–53](ch03.xhtml#page_52)

+= (addition shorthand), [66](ch03.xhtml#page_66)

\* (asterisk), importing all functions with, [245](ch11.xhtml#page_245)

/ (division operator), [62](ch03.xhtml#page_62)

/= (division shorthand), [66](ch03.xhtml#page_66)

"" (double quotation marks), for strings, [70](ch04.xhtml#page_70)

= (equal sign), assigning values to variables with, [32](ch02.xhtml#page_32)

== (equal to), [88](ch05.xhtml#page_88)

\*\* (exponential operator), [64–65](ch03.xhtml#page_64), [95–96](ch05.xhtml#page_95)

> (greater than), [92](ch05.xhtml#page_92)

>= (greater than or equal to), [93–94](ch05.xhtml#page_93)

# (hash mark), for comments, [39](ch02.xhtml#page_39)

< (less than), [92–93](ch05.xhtml#page_92)

<= (less than or equal to), [93–94](ch05.xhtml#page_93)

\* (multiplication operator), [62](ch03.xhtml#page_62)

\*= (multiplication shorthand), [66](ch03.xhtml#page_66)

!= (not equal to), [90–91](ch05.xhtml#page_90)

'' (single quotation marks), for strings, [70](ch04.xhtml#page_70)

[] (square brackets), for defining lists, [172](ch09.xhtml#page_172)

- (subtraction operator), [56](ch03.xhtml#page_56)

-= (subtraction shorthand), [66](ch03.xhtml#page_66)

""" (triple quotation marks), for docstrings, [156–157](ch08.xhtml#page_156)

2D lists, [212–217](ch10.xhtml#page_212), [220](ch10.xhtml#page_220)

3D lists, [222–229](ch10.xhtml#page_222)

### **A**

a (append permission), [237](ch11.xhtml#page_237)

addition operator (+), [52–53](ch03.xhtml#page_52)

shorthand (+=), [66](ch03.xhtml#page_66)

aliases, for modules, [245](ch11.xhtml#page_245)

and operator, [97](ch05.xhtml#page_97)

API (application programming interface), Minecraft Python

installing on Mac, [16–18](ch01.xhtml#page_16)

installing on Windows, [6–7](ch01.xhtml#page_6)

troubleshooting, [297–298](app02.xhtml#page_297)

append permission (a), [237](ch11.xhtml#page_237)

append() function, [175–176](ch09.xhtml#page_175)

application programming interface. *See* API

arguments, [38](ch02.xhtml#page_38), [151–152](ch08.xhtml#page_151)

line breaks in, [157](ch08.xhtml#page_157)

math operators in, [58–59](ch03.xhtml#page_58)

arrays. *See* lists

asterisk (\*), importing all functions with, [245](ch11.xhtml#page_245)

attributes, [261](ch12.xhtml#page_261). *See also* variables

accessing, [263–264](ch12.xhtml#page_263)

class, [275–277](ch12.xhtml#page_275)

### **B**

block hits program, [184–186](ch09.xhtml#page_184), [200–202](ch10.xhtml#page_200)

scoreboard, [196–198](ch09.xhtml#page_196), [209–210](ch10.xhtml#page_209)

blocks

changing, [56–57](ch03.xhtml#page_56), [142–143](ch07.xhtml#page_142), [200–202](ch10.xhtml#page_200)

finding highest, [94](ch05.xhtml#page_94)

identifying, [89](ch05.xhtml#page_89)

IDs

cheat sheet, [299–304](app03.xhtml#page_299)

finding by, [101–102](ch05.xhtml#page_101), [190–191](ch09.xhtml#page_190), [211–212](ch10.xhtml#page_211)

reminder program, [159–160](ch08.xhtml#page_159)

moving, [167–169](ch08.xhtml#page_167)

placing, [53](ch03.xhtml#page_53), [59–60](ch03.xhtml#page_59)

by user input, [78–79](ch04.xhtml#page_78)

random, [164–165](ch08.xhtml#page_164), [187](ch09.xhtml#page_187)

replacing, [177–178](ch09.xhtml#page_177)

stacking, [53–54](ch03.xhtml#page_53)

state, [162–163](ch08.xhtml#page_162)

wool, setting color by name, [162–163](ch08.xhtml#page_162)

Boolean operators. *See* logical operators

Boolean values, [86–87](ch05.xhtml#page_86)

break statements, [143](ch07.xhtml#page_143), [211](ch10.xhtml#page_211)

building quickly, [59–61](ch03.xhtml#page_59)

### **C**

chat

persistent, [143–144](ch07.xhtml#page_143)

posting to, [71–72](ch04.xhtml#page_71), [73–74](ch04.xhtml#page_73)

usernames, [76–77](ch04.xhtml#page_76)

cheat sheet, block IDs, [299–304](app03.xhtml#page_299)

choice() function, [186](ch09.xhtml#page_186)

class attributes, [275–277](ch12.xhtml#page_275). *See also* attributes, global variables

classes, [261–264](ch12.xhtml#page_261), [277–278](ch12.xhtml#page_277)

close() function, [237–238](ch11.xhtml#page_237), [251–252](ch11.xhtml#page_251)

color of wool blocks, setting by name, [162–163](ch08.xhtml#page_162)

command prompt, [24](ch01.xhtml#page_24), [27](ch01.xhtml#page_27)

comments, [39](ch02.xhtml#page_39), [156–157](ch08.xhtml#page_156)

comparators, [87–95](ch05.xhtml#page_87), [108–109](ch06.xhtml#page_108), [135–136](ch07.xhtml#page_135)

concatenation, [75–76](ch04.xhtml#page_75), [87](ch05.xhtml#page_87)

conditions, [85](ch05.xhtml#page_85), [108–109](ch06.xhtml#page_108), [135–136](ch07.xhtml#page_135)

connecting to Minecraft, [38](ch02.xhtml#page_38)

constructor, [263](ch12.xhtml#page_263)

coordinates, [35–36](ch02.xhtml#page_35)

copying structures, [229–233](ch10.xhtml#page_229), [246–250](ch11.xhtml#page_246), [252–256](ch11.xhtml#page_252)

count variables, [128](ch07.xhtml#page_128), [131–132](ch07.xhtml#page_131)

crater program, [109–110](ch06.xhtml#page_109)

curse program, [132–133](ch07.xhtml#page_132)

### **D**

dance floor, generating, [139–141](ch07.xhtml#page_139)

data. *See also* files

storing with variables, [32](ch02.xhtml#page_32)

types, [35](ch02.xhtml#page_35)

debugging, [46–48](ch02.xhtml#page_46)

decimal values, [41–42](ch02.xhtml#page_41)

decrementing values, [132](ch07.xhtml#page_132)

def keyword, [150](ch08.xhtml#page_150)

del keyword, [176–177](ch09.xhtml#page_176)

delays, setting in programs, [43–44](ch02.xhtml#page_43)

dictionaries. *See also* shelve module

defining, [192–193](ch09.xhtml#page_192)

items

accessing, [193](ch09.xhtml#page_193)

adding, [195–196](ch09.xhtml#page_195)

changing, [195–196](ch09.xhtml#page_195)

deleting, [196](ch09.xhtml#page_196)

looping over, [209](ch10.xhtml#page_209)

readability, [209](ch10.xhtml#page_209)

diving contest program, [136–138](ch07.xhtml#page_136)

division operator (/), [62](ch03.xhtml#page_62)

shorthand (/=), [66](ch03.xhtml#page_66)

docstrings, [156–157](ch08.xhtml#page_156)

double quotation marks (""), for strings, [70](ch04.xhtml#page_70)

dump() function, [243–244](ch11.xhtml#page_243)

### **E**

elif statements, [113–114](ch06.xhtml#page_113), [116–117](ch06.xhtml#page_116)

else statements, [111](ch06.xhtml#page_111), [145](ch07.xhtml#page_145), [210–211](ch10.xhtml#page_210)

else-if statements. *See* elif statements

equal to (==), [88](ch05.xhtml#page_88)

equal sign (=), assigning values to variables with, [32](ch02.xhtml#page_32)

errors. *See also* troubleshooting

debugging, [46–48](ch02.xhtml#page_46)

handling, [80–82](ch04.xhtml#page_80)

index, [172–173](ch09.xhtml#page_172)

scope, [166](ch08.xhtml#page_166)

syntax, [34](ch02.xhtml#page_34)

type, [151–152](ch08.xhtml#page_151), [158–159](ch08.xhtml#page_158)

exception handling, [80–82](ch04.xhtml#page_80)

exponential operator (\*\*), [64–65](ch03.xhtml#page_64), [95–96](ch05.xhtml#page_95)

expressions, [51–52](ch03.xhtml#page_51)

### **F**

False (Boolean value), [86](ch05.xhtml#page_86)

files, [235–239](ch11.xhtml#page_235)

opening, [236–237](ch11.xhtml#page_236), [251](ch11.xhtml#page_251)

reading, [238–239](ch11.xhtml#page_238)

saving, [237–238](ch11.xhtml#page_237), [251–252](ch11.xhtml#page_251)

shelve module, using with, [251–252](ch11.xhtml#page_251)

writing to, [237–238](ch11.xhtml#page_237)

Flask module, [257–259](ch11.xhtml#page_257). *See also* modules, pip

floats, [41–42](ch02.xhtml#page_41)

converting to strings, [75–76](ch04.xhtml#page_75)

flower trail, creating, [134–135](ch07.xhtml#page_134)

forest, building, [152–154](ch08.xhtml#page_152)

for loops, [199–200](ch10.xhtml#page_199)

with dictionaries, [209](ch10.xhtml#page_209)

generating 2D lists with, [220](ch10.xhtml#page_220)

with multidimensional lists, [212–217](ch10.xhtml#page_212), [222–229](ch10.xhtml#page_222)

for-else loops, [210–211](ch10.xhtml#page_210)

functions, [149](ch08.xhtml#page_149). *See also* methods

arguments, [151–152](ch08.xhtml#page_151)

calling, [150–151](ch08.xhtml#page_150)

defining, [150](ch08.xhtml#page_150)

returning values with, [157–159](ch08.xhtml#page_157), [183](ch09.xhtml#page_183), [270](ch12.xhtml#page_270)

### **G**

getBlock() function, [89](ch05.xhtml#page_89)

getHeight() function, [94](ch05.xhtml#page_94)

getPos() function, [60](ch03.xhtml#page_60)

getTilePos() function, [55](ch03.xhtml#page_55)

ghost structures

castle, [270–272](ch12.xhtml#page_270)

hotel, [279–281](ch12.xhtml#page_279)

house, [267–269](ch12.xhtml#page_267)

tree, [284–285](ch12.xhtml#page_284)

village, [273–275](ch12.xhtml#page_273)

gifts program, [114–115](ch06.xhtml#page_114)

global variables, [166–167](ch08.xhtml#page_166)

greater than (>), [92](ch05.xhtml#page_92)

greater than or equal to (>=), [93–94](ch05.xhtml#page_93)

### **H**

hardcoded values, [72](ch04.xhtml#page_72)

hash marks (#), for comments, [39](ch02.xhtml#page_39)

"Hello, Minecraft World", posting to chat, [71–72](ch04.xhtml#page_71)

hot and cold game, [145–147](ch07.xhtml#page_145)

### **I**

IDLE, [23–27](ch01.xhtml#page_23)

if statements, [107–109](ch06.xhtml#page_107)

with Boolean operators, [123–124](ch06.xhtml#page_123)

in functions, [161–162](ch08.xhtml#page_161)

with lists, [189–190](ch09.xhtml#page_189)

nested, [119](ch06.xhtml#page_119), [141](ch07.xhtml#page_141)

with range checks, [121](ch06.xhtml#page_121)

importing modules, [43–44](ch02.xhtml#page_43), [242–245](ch11.xhtml#page_242)

immutable

strings, [179](ch09.xhtml#page_179)

world, [86–87](ch05.xhtml#page_86), [112–113](ch06.xhtml#page_112)

in operator, [189–190](ch09.xhtml#page_189)

increment, [131–132](ch07.xhtml#page_131)

indentation, [80](ch04.xhtml#page_80), [108](ch06.xhtml#page_108), [150](ch08.xhtml#page_150)

index, of a list, [172–173](ch09.xhtml#page_172), [217–218](ch10.xhtml#page_217), [227–229](ch10.xhtml#page_227)

infinite loops, [131–132](ch07.xhtml#page_131)

inheritance, [277–279](ch12.xhtml#page_277), [282–284](ch12.xhtml#page_282)

\_\_init\_\_() method, [262–264](ch12.xhtml#page_262), [282–284](ch12.xhtml#page_282)

input

numbers only, [81–82](ch04.xhtml#page_81)

placing blocks by, [78–79](ch04.xhtml#page_78)

input() function, [72–73](ch04.xhtml#page_72)

installation. *See* Mac, Raspberry Pi, Windows

int() function, [78](ch04.xhtml#page_78)

integers, [35](ch02.xhtml#page_35)

converting to a string, [75–76](ch04.xhtml#page_75)

range checks, [121](ch06.xhtml#page_121), [139](ch07.xhtml#page_139)

iteration, [127–128](ch07.xhtml#page_127)

### **J**

Java

installing on Mac, [15–16](ch01.xhtml#page_15)

installing on Windows, [4–5](ch01.xhtml#page_4)

troubleshooting, [294](app02.xhtml#page_294), [296](app02.xhtml#page_296)

joining strings, [75–76](ch04.xhtml#page_75)

### **K**

keys, in dictionaries, [192–193](ch09.xhtml#page_192)

### **L**

lava trap, setting, [56–57](ch03.xhtml#page_56)

len() function, [183](ch09.xhtml#page_183)

less than (<), [92–93](ch05.xhtml#page_92)

less than or equal to (<=), [93–94](ch05.xhtml#page_93)

lists, [171–173](ch09.xhtml#page_171), [212–217](ch10.xhtml#page_212)

copying, [187–189](ch09.xhtml#page_187)

creating, [172](ch09.xhtml#page_172)

generating with range(), [202–203](ch10.xhtml#page_202), [204–205](ch10.xhtml#page_204)

index positions of, [172–173](ch09.xhtml#page_172), [217–218](ch10.xhtml#page_217), [227–229](ch10.xhtml#page_227)

items in

accessing, [172–173](ch09.xhtml#page_172)

adding, [175–176](ch09.xhtml#page_175)

changing, [173](ch09.xhtml#page_173)

deleting, [176–177](ch09.xhtml#page_176)

finding, [189–190](ch09.xhtml#page_189)

inserting, [176](ch09.xhtml#page_176)

length, [183](ch09.xhtml#page_183)

slicing, [188–189](ch09.xhtml#page_188)

three-dimensional, [222–229](ch10.xhtml#page_222)

two-dimensional, [212–217](ch10.xhtml#page_212), [220](ch10.xhtml#page_220)

list slice, [188–189](ch09.xhtml#page_188)

list() function, [204–205](ch10.xhtml#page_204)

load() function, [245](ch11.xhtml#page_245)

local variables, [166–167](ch08.xhtml#page_166)

logical operators, [96–104](ch05.xhtml#page_96)

and, [97](ch05.xhtml#page_97)

and if statements, [123–124](ch06.xhtml#page_123)

not, [100–101](ch05.xhtml#page_100)

or, [99](ch05.xhtml#page_99)

order of operations, [102–103](ch05.xhtml#page_102)

and while loops, [138–139](ch07.xhtml#page_138)

loops. *See* for loops, while loops

### **M**

Mac, setup instructions, [13–22](ch01.xhtml#page_13)

magic wand program, [200–202](ch10.xhtml#page_200)

math module, [146](ch07.xhtml#page_146)

math operators, [52–62](ch03.xhtml#page_52)

addition (+), [52–53](ch03.xhtml#page_52)

exponential (\*\*), [64–65](ch03.xhtml#page_64), [95–96](ch05.xhtml#page_95)

division (/), [62](ch03.xhtml#page_62)

multiplication (\*), [62](ch03.xhtml#page_62)

order of operations, [65](ch03.xhtml#page_65)

shorthand, [66](ch03.xhtml#page_66)

subtraction (-), [56](ch03.xhtml#page_56)

methods, [261](ch12.xhtml#page_261), [265–267](ch12.xhtml#page_265). *See also* classes, functions

adding to subclasses, [279](ch12.xhtml#page_279)

inheritance, [278](ch12.xhtml#page_278)

overriding, [282–284](ch12.xhtml#page_282)

returning values with, [270](ch12.xhtml#page_270)

Midas touch program, [142–143](ch07.xhtml#page_142)

Minecraft

API (application programming interface)

installing on Mac, [16–18](ch01.xhtml#page_16)

installing on Windows, [6–7](ch01.xhtml#page_6)

connecting programs to, [38](ch02.xhtml#page_38)

game

installing on Mac, [13–14](ch01.xhtml#page_13)

installing on Windows, [2–3](ch01.xhtml#page_2)

playing offline

on Mac, [21](ch01.xhtml#page_21)

on Windows, [11](ch01.xhtml#page_11)

profile

creating on Mac, [18–20](ch01.xhtml#page_18)

creating on Windows, [7–9](ch01.xhtml#page_7)

server

installing on Mac, [16–18](ch01.xhtml#page_16)

installing on Windows, [6–7](ch01.xhtml#page_6)

troubleshooting, [290–293](app02.xhtml#page_290)

worlds, creating new

on Mac, [18–21](ch01.xhtml#page_18)

on Windows, [7–11](ch01.xhtml#page_7)

modules, [242–245](ch11.xhtml#page_242)

installing with pip, [256–257](ch11.xhtml#page_256)

nicknames for, [245](ch11.xhtml#page_245)

pickle, [242–245](ch11.xhtml#page_242)

shelve, [251–252](ch11.xhtml#page_251)

time, [43–44](ch02.xhtml#page_43)

moving block program, [167–169](ch08.xhtml#page_167)

multiplication operator (\*), [62](ch03.xhtml#page_62)

shorthand (\*=), [66](ch03.xhtml#page_66)

### **N**

nicknames, for modules, [245](ch11.xhtml#page_245)

night vision sword program, [190–191](ch09.xhtml#page_190)

not equal to (!=), [90–91](ch05.xhtml#page_90)

not operator, [100–101](ch05.xhtml#page_100)

### **O**

object-oriented programming, [261–262](ch12.xhtml#page_261)

objects, [261–264](ch12.xhtml#page_261), [273–274](ch12.xhtml#page_273)

offline, playing Minecraft

on Mac, [21](ch01.xhtml#page_21)

on Windows, [11](ch01.xhtml#page_11)

open() function, [236–237](ch11.xhtml#page_236), [251](ch11.xhtml#page_251)

operators. *See* logical operators, math operators

or operator, [99](ch05.xhtml#page_99)

order of operations

logical operators, [102–103](ch05.xhtml#page_102)

math operators, [65](ch03.xhtml#page_65)

OS X, setup instructions, [13–22](ch01.xhtml#page_13)

### **P**

package manager, [256](ch11.xhtml#page_256)

parameters, of functions, [152](ch08.xhtml#page_152)

permissions, for files, [236–237](ch11.xhtml#page_236), [243](ch11.xhtml#page_243)

pickle module, [242–245](ch11.xhtml#page_242)

pillars, building, [206–207](ch10.xhtml#page_206)

pip, installing modules with, [256–257](ch11.xhtml#page_256)

pixel art, [218–219](ch10.xhtml#page_218)

pollBlockHits() function, [184–186](ch09.xhtml#page_184), [200–202](ch10.xhtml#page_200)

position, of player, [35–37](ch02.xhtml#page_35). *See also* teleporting

changing, [38](ch02.xhtml#page_38)

finding, [55](ch03.xhtml#page_55), [60](ch03.xhtml#page_60)

in specific environments, [89–90](ch05.xhtml#page_89), [91–92](ch05.xhtml#page_91), [94–95](ch05.xhtml#page_94), [97–100](ch05.xhtml#page_97)

in specific locations, [95–96](ch05.xhtml#page_95), [104–105](ch05.xhtml#page_104)

highest and lowest, [173–175](ch09.xhtml#page_173)

postToChat() function, [71–72](ch04.xhtml#page_71)

print() function, [70–71](ch04.xhtml#page_70)

progress bar, [177–178](ch09.xhtml#page_177)

pyramid, building, [207–208](ch10.xhtml#page_207)

Python

installing on Mac, [15](ch01.xhtml#page_15)

installing on Windows, [3–4](ch01.xhtml#page_3)

troubleshooting, [293](app02.xhtml#page_293), [294–295](app02.xhtml#page_294)

Python shell, [24](ch01.xhtml#page_24), [26–27](ch01.xhtml#page_26)

### **Q**

quotation marks

for docstrings, [156–157](ch08.xhtml#page_156)

for strings, [70](ch04.xhtml#page_70)

### **R**

r (read permission), [237](ch11.xhtml#page_237)

r+ (read-and-write permission), [237](ch11.xhtml#page_237)

randint() function, [66–67](ch03.xhtml#page_66)

random module, [66–67](ch03.xhtml#page_66), [186–187](ch09.xhtml#page_186)

range checks, [121](ch06.xhtml#page_121), [139](ch07.xhtml#page_139)

range() function, [202–203](ch10.xhtml#page_202), [204–205](ch10.xhtml#page_204)

Raspberry Pi, setup instructions, [22–23](ch01.xhtml#page_22)

read-and-write permission (r+), [237](ch11.xhtml#page_237)

read permission (r), [237](ch11.xhtml#page_237)

read() function, [238](ch11.xhtml#page_238)

readline() function, [238–239](ch11.xhtml#page_238)

refactoring, [154–156](ch08.xhtml#page_154)

return keyword, [157–159](ch08.xhtml#page_157), [183](ch09.xhtml#page_183), [270](ch12.xhtml#page_270)

reversed() function, [205–206](ch10.xhtml#page_205)

running a program, [40](ch02.xhtml#page_40)

### **S**

scope, of variables, [166–167](ch08.xhtml#page_166)

scoreboard, for block hits game, [196–198](ch09.xhtml#page_196), [209–210](ch10.xhtml#page_209)

secret passage, building, [119–120](ch06.xhtml#page_119)

server

installing on Mac, [16–18](ch01.xhtml#page_16)

installing on Windows, [6–7](ch01.xhtml#page_6)

troubleshooting, [296](app02.xhtml#page_296)

setBlock() function, [53](ch03.xhtml#page_53), [162–163](ch08.xhtml#page_162)

setBlocks() function, [59–60](ch03.xhtml#page_59)

setPos() function, [42](ch02.xhtml#page_42)

setTilePos() function, [38–39](ch02.xhtml#page_38)

setting() function, [86–87](ch05.xhtml#page_86)

setup instructions

for Mac, [13–22](ch01.xhtml#page_13)

for Raspberry Pi, [22–23](ch01.xhtml#page_22)

for Windows, [2–12](ch01.xhtml#page_2)

shell, [24](ch01.xhtml#page_24), [26–27](ch01.xhtml#page_26)

shelve module, [251–252](ch11.xhtml#page_251)

shorthand operators, [66](ch03.xhtml#page_66)

shower program, [124–126](ch06.xhtml#page_124)

sightseeing guide, creating, [194–195](ch09.xhtml#page_194)

single quotation marks (''), for strings, [70](ch04.xhtml#page_70)

sleep() function, [43–44](ch02.xhtml#page_43)

slices, of lists, [188–189](ch09.xhtml#page_188)

sliding program, [181–182](ch09.xhtml#page_181)

smashing, preventing, [86–87](ch05.xhtml#page_86), [112–113](ch06.xhtml#page_112)

Spigot

on Mac, [16–22](ch01.xhtml#page_16)

on Windows, [6–12](ch01.xhtml#page_6)

spires, creating, [62–64](ch03.xhtml#page_62)

sprint record, [82–84](ch04.xhtml#page_82)

sqrt() function, [146](ch07.xhtml#page_146)

square brackets ([]), for defining lists, [172](ch09.xhtml#page_172)

square root, calculating, [146](ch07.xhtml#page_146)

stairs, building, [203–204](ch10.xhtml#page_203)

state, of blocks, [162–163](ch08.xhtml#page_162)

statements, [33–34](ch02.xhtml#page_33), [51–52](ch03.xhtml#page_51)

str() function, [75–76](ch04.xhtml#page_75), [87](ch05.xhtml#page_87)

strings, [70](ch04.xhtml#page_70)

accessing characters in, [179](ch09.xhtml#page_179)

concatenating, [75–76](ch04.xhtml#page_75)

converting to integers, [78](ch04.xhtml#page_78)

subclasses, [277–279](ch12.xhtml#page_277), [282–284](ch12.xhtml#page_282)

subtraction operator (-), [56](ch03.xhtml#page_56)

shorthand (-=), [66](ch03.xhtml#page_66)

super jump program, [67–68](ch03.xhtml#page_67)

superclasses, [277–279](ch12.xhtml#page_277), [282–284](ch12.xhtml#page_282)

survival mode

on Mac, [21–22](ch01.xhtml#page_21)

on Windows, [11–12](ch01.xhtml#page_11)

sword

hits, [184–186](ch09.xhtml#page_184), [200–202](ch10.xhtml#page_200)

magic wand, [200–202](ch10.xhtml#page_200)

night vision, [190–191](ch09.xhtml#page_190)

syntax, [33–34](ch02.xhtml#page_33)

### **T**

teleporting, [35–39](ch02.xhtml#page_35), [44–46](ch02.xhtml#page_44)

by location name, [194–195](ch09.xhtml#page_194), [264–265](ch12.xhtml#page_264)

by point score, [117–118](ch06.xhtml#page_117)

precisely, [42–43](ch02.xhtml#page_42)

to random locations, [129–130](ch07.xhtml#page_129)

restrictions, [122–123](ch06.xhtml#page_122)

text. *See* files, strings

text editor, [24–26](ch01.xhtml#page_24)

three-dimensional lists, [222–229](ch10.xhtml#page_222)

throwing an exception, [80](ch04.xhtml#page_80)

time module, [43–44](ch02.xhtml#page_43)

to-do list, [239–241](ch11.xhtml#page_239)

triple quotation marks ("""), for docstrings, [156–157](ch08.xhtml#page_156)

troubleshooting, [289–298](app02.xhtml#page_289)

API, [297–298](app02.xhtml#page_297)

Java, [294](app02.xhtml#page_294), [296](app02.xhtml#page_296)

Minecraft, [290–293](app02.xhtml#page_290)

Python, [293](app02.xhtml#page_293), [294–295](app02.xhtml#page_294)

server, [296](app02.xhtml#page_296)

True (Boolean value), [86](ch05.xhtml#page_86)

try-except statements, [80–82](ch04.xhtml#page_80)

tuples, [179–180](ch09.xhtml#page_179), [183](ch09.xhtml#page_183)

two-dimensional lists, [212–217](ch10.xhtml#page_212), [220](ch10.xhtml#page_220)

TypeError, [151–152](ch08.xhtml#page_151), [158–159](ch08.xhtml#page_158)

### **U**

UnboundLocalError, [166](ch08.xhtml#page_166)

usernames, adding to chat, [76–77](ch04.xhtml#page_76)

### **V**

values

in dictionaries, [192–193](ch09.xhtml#page_192), [195–196](ch09.xhtml#page_195)

of variables, [32](ch02.xhtml#page_32)

variables, [32–35](ch02.xhtml#page_32), [172](ch09.xhtml#page_172)

assigning values to, [32](ch02.xhtml#page_32)

changing values of, [35](ch02.xhtml#page_35)

global, [166–167](ch08.xhtml#page_166)

local, [166–167](ch08.xhtml#page_166)

naming, [32–33](ch02.xhtml#page_32)

syntax, [33–34](ch02.xhtml#page_33)

### **W**

w (write permission), [236–237](ch11.xhtml#page_236)

waiting, in programs, [43–44](ch02.xhtml#page_43)

wand, magic, [200–202](ch10.xhtml#page_200)

watery curse program, [132–133](ch07.xhtml#page_132)

weather-worn wall, building, [221–222](ch10.xhtml#page_221)

website, creating with Flask, [257–259](ch11.xhtml#page_257)

while loops, [127–128](ch07.xhtml#page_127)

conditions, [135–136](ch07.xhtml#page_135)

ending, [131–132](ch07.xhtml#page_131), [143](ch07.xhtml#page_143)

with if statements, [141](ch07.xhtml#page_141)

infinite, [131–132](ch07.xhtml#page_131), [134](ch07.xhtml#page_134)

with return statements, [164](ch08.xhtml#page_164)

while-else statements, [145](ch07.xhtml#page_145)

Windows, setup instructions, [2–12](ch01.xhtml#page_2)

wool blocks, setting color by name, [162–163](ch08.xhtml#page_162)

worlds (Minecraft), creating new

on Mac, [18–21](ch01.xhtml#page_18)

on Windows, [7–11](ch01.xhtml#page_7)

write permission (w), [236–237](ch11.xhtml#page_236)

write() function, [237–238](ch11.xhtml#page_237)

### **X**

x, y, and z coordinates, [35–36](ch02.xhtml#page_35)