

# CONTENTS IN DETAIL

## ACKNOWLEDGMENTS

XVII

## INTRODUCTION

XIX

Why Learn to Program? . . . . .	xx
Why Python? . . . . .	xx
Why Minecraft? . . . . .	xx
What's in This Book? . . . . .	xx
Online Resources . . . . .	xxii
Let the Adventure Begin! . . . . .	xxii

## 1

## SETTING UP FOR YOUR ADVENTURE

1

Setting Up Your Windows PC . . . . .	2
Installing Minecraft . . . . .	2
Installing Python . . . . .	3
Installing Java . . . . .	4
Installing the Minecraft Python API and Spigot . . . . .	6
Running Spigot and Creating a Game . . . . .	7
Starting Over with a New World . . . . .	8
Playing Offline . . . . .	9
Switching to Survival Mode . . . . .	10
Setting Up Your Mac . . . . .	11
Installing Minecraft . . . . .	12
Installing Python . . . . .	13
Installing Java . . . . .	14
Installing the Minecraft Python API and Spigot . . . . .	15
Running Spigot and Creating a Game . . . . .	16
Starting Over with a New World . . . . .	17
Playing Offline . . . . .	18
Switching to Survival Mode . . . . .	18
Setting Up Your Raspberry Pi . . . . .	18
Getting to Know IDLE . . . . .	20
Getting to Know the Python Shell . . . . .	20
Say Hello to IDLE's Text Editor . . . . .	21
When to Use the Python Shell and When to Use the Text Editor . . . . .	23
The Prompts Used in This Book . . . . .	23
Testing Your Minecraft Python Setup . . . . .	24

<b>2</b>	<b>TELEPORTING WITH VARIABLES</b>	<b>27</b>
What Is a Program? . . . . .		27
Storing Data with Variables. . . . .		28
The Structure of Programming Languages. . . . .		29
Syntax Rules for Variables . . . . .		30
Changing the Values of Variables. . . . .		30
Integers . . . . .		31
<b>Mission #1: Teleport the Player</b> . . . . .		<b>31</b>
Floats . . . . .		37
<b>Mission #2: Go Exactly Where You Want.</b> . . . . .		<b>38</b>
Slowing Down Teleportation Using the time Module . . . . .		39
<b>Mission #3: Teleportation Tour</b> . . . . .		<b>40</b>
Debugging . . . . .		42
<b>Mission #4: Fix the Buggy Teleportation.</b> . . . . .		<b>43</b>
What You Learned . . . . .		45
<b>3</b>	<b>BUILDING QUICKLY AND TRAVELING FAR WITH MATH</b>	<b>47</b>
Expressions and Statements. . . . .		47
Operators . . . . .		48
Addition . . . . .		48
<b>Mission #5: Stack Blocks</b> . . . . .		<b>49</b>
<b>Mission #6: Super Jump</b> . . . . .		<b>51</b>
Subtraction . . . . .		52
<b>Mission #7: Change the Blocks Under You</b> . . . . .		<b>52</b>
Using Math Operators in Arguments . . . . .		54
<b>Mission #8: Speed Building</b> . . . . .		<b>55</b>
Multiplication . . . . .		58
Division . . . . .		58
<b>Mission #9: Spectacular Spires</b> . . . . .		<b>58</b>
Exponents . . . . .		60
Parentheses and Order of Operations . . . . .		61
Handy Math Tricks . . . . .		62
Shorthand Operators. . . . .		62
Playing with Random Numbers . . . . .		62
<b>Mission #10: Super Jump Somewhere New!</b> . . . . .		<b>63</b>
What You Learned . . . . .		64
<b>4</b>	<b>CHATting WITH STRINGS</b>	<b>65</b>
What Are Strings? . . . . .		66
The print() Function. . . . .		66
<b>Mission #11: Hello, Minecraft World</b> . . . . .		<b>67</b>
The input() Function . . . . .		68
<b>Mission #12: Write Your Own Chat Message.</b> . . . . .		<b>69</b>
Joining Strings . . . . .		71
Converting Numbers to Strings . . . . .		71
Concatenating Integers and Floats. . . . .		72
<b>Mission #13: Add Usernames to Chat</b> . . . . .		<b>72</b>

Converting Strings to Integers with int() . . . . .	74
<b>Mission #14: Create a Block with input</b> . . . . .	<b>74</b>
Bounce Back from Errors . . . . .	76
<b>Mission #15: Only Numbers Allowed</b> . . . . .	<b>77</b>
<b>Mission #16: Sprint Record</b> . . . . .	<b>78</b>
What You Learned . . . . .	80

## 5 FIGURING OUT WHAT'S TRUE AND FALSE WITH BOOLEANS 81

Boolean Basics . . . . .	82
<b>Mission #17: Stop Smashing Blocks!</b> . . . . .	<b>82</b>
Concatenating Booleans . . . . .	83
Comparators . . . . .	83
Equal To . . . . .	84
<b>Mission #18: Am I Swimming?</b> . . . . .	<b>85</b>
Not Equal To . . . . .	86
<b>Mission #19: Am I Standing in Something Other Than Air?</b> . . . . .	<b>87</b>
Greater Than and Less Than . . . . .	88
Greater Than or Equal To and Less Than or Equal To . . . . .	89
<b>Mission #20: Am I Above the Ground?</b> . . . . .	<b>90</b>
<b>Mission #21: Am I Close to Home?</b> . . . . .	<b>91</b>
Logical Operators . . . . .	92
and . . . . .	93
<b>Mission #22: Am I Entirely Underwater?</b> . . . . .	<b>93</b>
or . . . . .	95
<b>Mission #23: Am I in a Tree?</b> . . . . .	<b>95</b>
not . . . . .	96
<b>Mission #24: Is This Block Not a Melon?</b> . . . . .	<b>97</b>
Logical Operator Order . . . . .	98
Is My Number Between Two Others? . . . . .	99
<b>Mission #25: Am I in the House?</b> . . . . .	<b>100</b>
What You Learned . . . . .	101

## 6 MAKING MINI-GAMES WITH IF STATEMENTS 103

Using if Statements . . . . .	104
<b>Mission #26: Blast a Crater</b> . . . . .	<b>105</b>
else Statements . . . . .	107
<b>Mission #27: Prevent Smashing, or Not</b> . . . . .	<b>108</b>
elif Statements . . . . .	109
<b>Mission #28: Offer a Gift</b> . . . . .	<b>110</b>
Chaining Together elif Statements . . . . .	112
<b>Mission #29: Teleport to the Right Place.</b> . . . . .	<b>113</b>
Nested if Statements . . . . .	115
<b>Mission #30: Open a Secret Passage.</b> . . . . .	<b>115</b>
Using if Statements to Test a Range of Values . . . . .	117
<b>Mission #31: Restrict Teleport Locations</b> . . . . .	<b>118</b>
Boolean Operators and if Statements . . . . .	119
<b>Mission #32: Take a Shower</b> . . . . .	<b>120</b>
What You Learned . . . . .	122

## 7 DANCE PARTIES AND FLOWER PARADES WITH WHILE LOOPS 123

A Simple while Loop . . . . .	123
<b>Mission #33: A Random Teleportation Tour</b> . . . . .	<b>125</b>
Controlling Loops with a Count Variable . . . . .	127
<b>Mission #34: The Watery Curse</b> . . . . .	<b>128</b>
Infinite while Loops . . . . .	130
<b>Mission #35: Flower Trail</b> . . . . .	<b>130</b>
Fancy Conditions . . . . .	131
<b>Mission #36: Diving Contest</b> . . . . .	<b>132</b>
Boolean Operators and while Loops . . . . .	134
Checking a Range of Values in while Loops . . . . .	135
<b>Mission #37: Make a Dance Floor</b> . . . . .	<b>135</b>
Nested if Statements and while Loops . . . . .	137
<b>Mission #38: The Midas Touch</b> . . . . .	<b>138</b>
Ending a while Loop with break . . . . .	139
<b>Mission #39: Create a Persistent Chat with a Loop</b> . . . . .	<b>139</b>
while-else Statements . . . . .	141
<b>Mission #40: Hot and Cold</b> . . . . .	<b>141</b>
What You Learned . . . . .	144

## 8 FUNCTIONS GIVE YOU SUPERPOWERS 145

Defining Your Own Functions . . . . .	146
Calling a Function . . . . .	146
Functions Take Arguments . . . . .	147
<b>Mission #41: Build a Forest</b> . . . . .	<b>148</b>
Refactoring a Program . . . . .	150
<b>Mission #42: Refactor Away</b> . . . . .	<b>151</b>
Commenting with Docstrings . . . . .	152
Line Breaks in Arguments . . . . .	153
Function Return Values . . . . .	153
<b>Mission #43: Block ID Reminder</b> . . . . .	<b>155</b>
Using if Statements and while Loops in Functions . . . . .	157
if Statements . . . . .	157
<b>Mission #44: Wool Color Helper</b> . . . . .	<b>158</b>
while Loops . . . . .	159
<b>Mission #45: Blocks, Everywhere</b> . . . . .	<b>160</b>
Global and Local Variables . . . . .	162
<b>Mission #46: A Moving Block</b> . . . . .	<b>163</b>
What You Learned . . . . .	165

## 9 HITTING THINGS WITH LISTS AND DICTIONARIES 167

Using Lists . . . . .	168
Accessing a List Item . . . . .	168
Changing a List Item . . . . .	169
<b>Mission #47: High and Low</b> . . . . .	<b>169</b>

Manipulating Lists . . . . .	171
Adding an Item . . . . .	171
Inserting an Item . . . . .	172
Deleting an Item . . . . .	172
<b>Mission #48: Progress Bar</b> . . . . .	<b>173</b>
Treating Strings like Lists . . . . .	175
Tuples . . . . .	175
Setting Variables with Tuples . . . . .	176
<b>Mission #49: Sliding</b> . . . . .	<b>177</b>
Returning a Tuple . . . . .	179
Other Useful Features of Lists . . . . .	179
List Length . . . . .	179
<b>Mission #50: Block Hits</b> . . . . .	<b>180</b>
Randomly Choosing an Item . . . . .	182
<b>Mission #51: Random Block</b> . . . . .	<b>183</b>
Copying a List . . . . .	183
Items and if Statements . . . . .	185
<b>Mission #52: Night Vision Sword</b> . . . . .	<b>186</b>
Dictionaries . . . . .	188
Defining a Dictionary . . . . .	188
Accessing Items in Dictionaries . . . . .	189
<b>Mission #53: Sightseeing Guide</b> . . . . .	<b>190</b>
Changing or Adding an Item in a Dictionary . . . . .	191
Deleting Items in Dictionaries . . . . .	192
<b>Mission #54: Block Hits Score</b> . . . . .	<b>192</b>
What You Learned . . . . .	194

## 10 MINECRAFT MAGIC WITH FOR LOOPS 195

A Simple for Loop . . . . .	195
<b>Mission #55: Magic Wand</b> . . . . .	<b>196</b>
The range() Function . . . . .	198
<b>Mission #56: Magic Stairs</b> . . . . .	<b>199</b>
Playing Around with range() . . . . .	200
Other List Functions . . . . .	201
<b>Mission #57: Pillars</b> . . . . .	<b>202</b>
<b>Mission #58: Pyramid</b> . . . . .	<b>203</b>
Looping Over a Dictionary . . . . .	205
<b>Mission #59: Scoreboard</b> . . . . .	<b>205</b>
for-else Loops . . . . .	206
Breaking a for-else Loop . . . . .	207
<b>Mission #60: The Diamond Prospector</b> . . . . .	<b>207</b>
Nested for Loops and Multidimensional Lists . . . . .	208
Thinking in Two Dimensions . . . . .	209
Accessing Values in 2D Lists . . . . .	213
<b>Mission #61: Pixel Art</b> . . . . .	<b>214</b>
Generating 2D Lists with Loops . . . . .	216
<b>Mission #62: A Weather-Worn Wall</b> . . . . .	<b>217</b>
Thinking in Three Dimensions . . . . .	218
Outputting 3D Lists . . . . .	219

Accessing Values in 3D Lists . . . . .	223
<b>Mission #63: Duplicate a Building</b> . . . . .	<b>225</b>
What You Learned . . . . .	230

## 11 SAVING AND LOADING BUILDINGS WITH FILES AND MODULES 231

Using Files . . . . .	232
Opening a File . . . . .	232
Writing to and Saving a File . . . . .	233
Reading a File . . . . .	234
Reading a Line of a File . . . . .	234
<b>Mission #64: To-Do List</b> . . . . .	<b>235</b>
Part 1: Writing the To-Do List . . . . .	235
Part 2: Displaying the To-Do List . . . . .	237
Using Modules. . . . .	238
The pickle Module. . . . .	238
Importing pickle . . . . .	238
Importing One Function with the from Clause . . . . .	240
Importing All Functions with * . . . . .	241
Giving a Module a Nickname . . . . .	241
<b>Mission #65: Save a Building</b> . . . . .	<b>242</b>
Part 1: Saving the Building . . . . .	242
Part 2: Loading the Building . . . . .	245
Storing Lots of Data with the shelve Module. . . . .	247
Opening a File with shelve. . . . .	247
Adding, Modifying, and Accessing Items with shelve. . . . .	247
<b>Mission #66: Save a Collection of Structures</b> . . . . .	<b>248</b>
Part 1: Saving a Structure to a Collection . . . . .	248
Part 2: Loading a Structure from a Collection . . . . .	249
Installing New Modules with pip . . . . .	252
Using pip on Windows . . . . .	252
Using pip on a Mac or Raspberry Pi . . . . .	253
Using a Module from pip: Flask. . . . .	253
<b>Mission #67: Position Website</b> . . . . .	<b>255</b>
What You Learned . . . . .	256

## 12 GETTING CLASSY WITH OBJECT-ORIENTED PROGRAMMING 257

Object-Oriented Basics . . . . .	258
Creating a Class . . . . .	258
Creating an Object . . . . .	259
Accessing Attributes . . . . .	259
<b>Mission #68: Location Objects</b> . . . . .	<b>260</b>
Understanding Methods . . . . .	261
<b>Mission #69: Ghost House</b> . . . . .	<b>263</b>
Returning Values with Methods . . . . .	266
<b>Mission #70: Ghost Castle</b> . . . . .	<b>266</b>

Creating Multiple Objects . . . . .	269
<b>Mission #71: Ghost Town</b> . . . . .	<b>269</b>
Class Attributes . . . . .	271
Understanding Inheritance. . . . .	273
Inheriting a Class . . . . .	274
Adding New Methods to Subclasses . . . . .	275
<b>Mission #72: Ghost Hotel</b> . . . . .	<b>275</b>
Overriding Methods and Attributes. . . . .	278
<b>Mission #73: Ghost Tree</b> . . . . .	<b>280</b>
What You Learned . . . . .	282

**AFTERWORD** **283**

**BLOCK ID CHEAT SHEET** **285**

**INDEX** **291**