

CONTENTS IN DETAIL

PREFACE TO THE SECOND EDITION	xxvii
ACKNOWLEDGMENTS	xxxi
INTRODUCTION	xxxiii
Who Is This Book For?	xxxiv
What Can You Expect to Learn?	xxxiv
Online Resources	xxxv
Why Python?	xxxvi

PART I: BASICS1

1	
GETTING STARTED	3
Setting Up Your Programming Environment	3
Python Versions	4
Running Snippets of Python Code	4
About the Sublime Text Editor	4
Python on Different Operating Systems	5
Python on Windows	5
Python on macOS	7
Python on Linux	8
Running a Hello World Program	9
Configuring Sublime Text to Use the Correct Python Version	9
Running hello_world.py	10
Troubleshooting	11
Running Python Programs from a Terminal	12
On Windows	12
On macOS and Linux	12
Exercise 1-1: python.org	13
Exercise 1-2: Hello World Typos	13
Exercise 1-3: Infinite Skills	13
Summary	13
2	
VARIABLES AND SIMPLE DATA TYPES	15
What Really Happens When You Run hello_world.py.	15
Variables	16
Naming and Using Variables	17
Avoiding Name Errors When Using Variables	17
Variables Are Labels	18
Exercise 2-1: Simple Message	19
Exercise 2-2: Simple Messages	19

Strings	19
Changing Case in a String with Methods	20
Using Variables in Strings	21
Adding Whitespace to Strings with Tabs or Newlines	22
Stripping Whitespace	22
Avoiding Syntax Errors with Strings	24
Exercise 2-3: Personal Message	25
Exercise 2-4: Name Cases	25
Exercise 2-5: Famous Quote	25
Exercise 2-6: Famous Quote 2	25
Exercise 2-7: Stripping Names	25
Numbers	25
Integers	26
Floats	26
Integers and Floats	27
Underscores in Numbers	28
Multiple Assignment	28
Constants	28
Exercise 2-8: Number Eight	29
Exercise 2-9: Favorite Number	29
Comments	29
How Do You Write Comments?	29
What Kind of Comments Should You Write?	29
Exercise 2-10: Adding Comments	30
The Zen of Python	30
Exercise 2-11: Zen of Python	31
Summary	32

3 INTRODUCING LISTS 33

What Is a List?	33
Accessing Elements in a List	34
Index Positions Start at 0, Not 1	35
Using Individual Values from a List	35
Exercise 3-1: Names	36
Exercise 3-2: Greetings	36
Exercise 3-3: Your Own List	36
Changing, Adding, and Removing Elements	36
Modifying Elements in a List	36
Adding Elements to a List	37
Removing Elements from a List	38
Exercise 3-4: Guest List	42
Exercise 3-5: Changing Guest List	42
Exercise 3-6: More Guests	42
Exercise 3-7: Shrinking Guest List	43
Organizing a List	43
Sorting a List Permanently with the sort() Method	43
Sorting a List Temporarily with the sorted() Function	44
Printing a List in Reverse Order	45
Finding the Length of a List	45
Exercise 3-8: Seeing the World	46
Exercise 3-9: Dinner Guests	46
Exercise 3-10: Every Function	46

Avoiding Index Errors When Working with Lists	46
Exercise 3-11: Intentional Error	48
Summary	48
4 WORKING WITH LISTS 49	
Looping Through an Entire List	49
A Closer Look at Looping	50
Doing More Work Within a for Loop	51
Doing Something After a for Loop	52
Avoiding Indentation Errors	53
Forgetting to Indent	53
Forgetting to Indent Additional Lines	54
Indenting Unnecessarily	55
Indenting Unnecessarily After the Loop	55
Forgetting the Colon	56
Exercise 4-1: Pizzas	56
Exercise 4-2: Animals	56
Making Numerical Lists	57
Using the range() Function	57
Using range() to Make a List of Numbers	58
Simple Statistics with a List of Numbers	59
List Comprehensions	59
Exercise 4-3: Counting to Twenty	60
Exercise 4-4: One Million	60
Exercise 4-5: Summing a Million	60
Exercise 4-6: Odd Numbers	60
Exercise 4-7: Threes	60
Exercise 4-8: Cubes	60
Exercise 4-9: Cube Comprehension	60
Working with Part of a List	61
Slicing a List	61
Looping Through a Slice	62
Copying a List	63
Exercise 4-10: Slices	65
Exercise 4-11: My Pizzas, Your Pizzas	65
Exercise 4-12: More Loops	65
Tuples	65
Defining a Tuple	66
Looping Through All Values in a Tuple	67
Writing over a Tuple	67
Exercise 4-13: Buffet	68
Styling Your Code	68
The Style Guide	68
Indentation	69
Line Length	69
Blank Lines	69
Other Style Guidelines	70
Exercise 4-14: PEP 8	70
Exercise 4-15: Code Review	70
Summary	70

5		
IF STATEMENTS		71
A Simple Example		72
Conditional Tests		72
Checking for Equality		72
Ignoring Case When Checking for Equality		73
Checking for Inequality		74
Numerical Comparisons		74
Checking Multiple Conditions		75
Checking Whether a Value Is in a List		76
Checking Whether a Value Is Not in a List		77
Boolean Expressions		77
<i>Exercise 5-1: Conditional Tests</i>		78
<i>Exercise 5-2: More Conditional Tests</i>		78
if Statements		78
Simple if Statements		78
if-else Statements		79
The if-elif-else Chain		80
Using Multiple elif Blocks		82
Omitting the else Block		82
Testing Multiple Conditions		83
<i>Exercise 5-3: Alien Colors #1</i>		84
<i>Exercise 5-4: Alien Colors #2</i>		84
<i>Exercise 5-5: Alien Colors #3</i>		85
<i>Exercise 5-6: Stages of Life</i>		85
<i>Exercise 5-7: Favorite Fruit</i>		85
Using if Statements with Lists		85
Checking for Special Items		86
Checking That a List Is Not Empty		87
Using Multiple Lists		88
<i>Exercise 5-8: Hello Admin</i>		89
<i>Exercise 5-9: No Users</i>		89
<i>Exercise 5-10: Checking Usernames</i>		89
<i>Exercise 5-11: Ordinal Numbers</i>		89
Styling Your if Statements		90
<i>Exercise 5-12: Styling if statements</i>		90
<i>Exercise 5-13: Your Ideas</i>		90
Summary		90
6		
DICTIONARIES		91
A Simple Dictionary		92
Working with Dictionaries		92
Accessing Values in a Dictionary		93
Adding New Key-Value Pairs		93
Starting with an Empty Dictionary		94
Modifying Values in a Dictionary		95
Removing Key-Value Pairs		96
A Dictionary of Similar Objects		97
Using <code>get()</code> to Access Values		98

<i>Exercise 6-1: Person</i>	99
<i>Exercise 6-2: Favorite Numbers</i>	99
<i>Exercise 6-3: Glossary</i>	99
Looping Through a Dictionary	99
Looping Through All Key-Value Pairs	99
Looping Through All the Keys in a Dictionary	101
Looping Through a Dictionary's Keys in a Particular Order	103
Looping Through All Values in a Dictionary	104
<i>Exercise 6-4: Glossary 2</i>	105
<i>Exercise 6-5: Rivers</i>	105
<i>Exercise 6-6: Polling</i>	105
Nesting	106
A List of Dictionaries	106
A List in a Dictionary	108
A Dictionary in a Dictionary	110
<i>Exercise 6-7: People</i>	112
<i>Exercise 6-8: Pets</i>	112
<i>Exercise 6-9: Favorite Places</i>	112
<i>Exercise 6-10: Favorite Numbers</i>	112
<i>Exercise 6-11: Cities</i>	112
<i>Exercise 6-12: Extensions</i>	112
Summary	112
7	
USER INPUT AND WHILE LOOPS	113
How the <code>input()</code> Function Works	114
Writing Clear Prompts	114
Using <code>int()</code> to Accept Numerical Input	115
The Modulo Operator	116
<i>Exercise 7-1: Rental Car</i>	117
<i>Exercise 7-2: Restaurant Seating</i>	117
<i>Exercise 7-3: Multiples of Ten</i>	117
Introducing while Loops	118
The while Loop in Action	118
Letting the User Choose When to Quit	118
Using a Flag	120
Using <code>break</code> to Exit a Loop	121
Using <code>continue</code> in a Loop	122
Avoiding Infinite Loops	122
<i>Exercise 7-4: Pizza Toppings</i>	123
<i>Exercise 7-5: Movie Tickets</i>	123
<i>Exercise 7-6: Three Exits</i>	124
<i>Exercise 7-7: Infinity</i>	124
Using a while Loop with Lists and Dictionaries	124
Moving Items from One List to Another	124
Removing All Instances of Specific Values from a List	125
Filling a Dictionary with User Input	126
<i>Exercise 7-8: Deli</i>	127
<i>Exercise 7-9: No Pastrami</i>	127
<i>Exercise 7-10: Dream Vacation</i>	127
Summary	127

8	FUNCTIONS	129
Defining a Function	130	
Passing Information to a Function	130	
Arguments and Parameters	131	
<i>Exercise 8-1: Message</i>	131	
<i>Exercise 8-2: Favorite Book</i>	131	
Passing Arguments	131	
Positional Arguments	132	
Keyword Arguments	133	
Default Values	134	
Equivalent Function Calls	135	
Avoiding Argument Errors	136	
<i>Exercise 8-3: T-Shirt</i>	137	
<i>Exercise 8-4: Large Shirts</i>	137	
<i>Exercise 8-5: Cities</i>	137	
Return Values	137	
Returning a Simple Value	138	
Making an Argument Optional	138	
Returning a Dictionary	140	
Using a Function with a while Loop	141	
<i>Exercise 8-6: City Names</i>	142	
<i>Exercise 8-7: Album</i>	142	
<i>Exercise 8-8: User Albums</i>	142	
Passing a List	143	
Modifying a List in a Function	143	
Preventing a Function from Modifying a List	145	
<i>Exercise 8-9: Messages</i>	146	
<i>Exercise 8-10: Sending Messages</i>	146	
<i>Exercise 8-11: Archived Messages</i>	146	
Passing an Arbitrary Number of Arguments	147	
Mixing Positional and Arbitrary Arguments	148	
Using Arbitrary Keyword Arguments	148	
<i>Exercise 8-12: Sandwiches</i>	150	
<i>Exercise 8-13: User Profile</i>	150	
<i>Exercise 8-14: Cars</i>	150	
Storing Your Functions in Modules	150	
Importing an Entire Module	150	
Importing Specific Functions	152	
Using as to Give a Function an Alias	152	
Using as to Give a Module an Alias	153	
Importing All Functions in a Module	153	
Styling Functions	154	
<i>Exercise 8-15: Printing Models</i>	155	
<i>Exercise 8-16: Imports</i>	155	
<i>Exercise 8-17: Styling Functions</i>	155	
Summary	155	

9	CLASSES	157
Creating and Using a Class	158	
Creating the Dog Class	158	
Making an Instance from a Class	160	
<i>Exercise 9-1: Restaurant</i>	162	
<i>Exercise 9-2: Three Restaurants</i>	162	
<i>Exercise 9-3: Users</i>	162	
Working with Classes and Instances	162	
The Car Class	162	
Setting a Default Value for an Attribute	163	
Modifying Attribute Values	164	
<i>Exercise 9-4: Number Served</i>	167	
<i>Exercise 9-5: Login Attempts</i>	167	
Inheritance	167	
The <code>__init__()</code> Method for a Child Class	167	
Defining Attributes and Methods for the Child Class	169	
Overriding Methods from the Parent Class	170	
Instances as Attributes	170	
Modeling Real-World Objects	173	
<i>Exercise 9-6: Ice Cream Stand</i>	173	
<i>Exercise 9-7: Admin</i>	173	
<i>Exercise 9-8: Privileges</i>	173	
<i>Exercise 9-9: Battery Upgrade</i>	174	
Importing Classes	174	
Importing a Single Class	174	
Storing Multiple Classes in a Module	175	
Importing Multiple Classes from a Module	177	
Importing an Entire Module	177	
Importing All Classes from a Module	177	
Importing a Module into a Module	178	
Using Aliases	179	
Finding Your Own Workflow	179	
<i>Exercise 9-10: Imported Restaurant</i>	180	
<i>Exercise 9-11: Imported Admin</i>	180	
<i>Exercise 9-12: Multiple Modules</i>	180	
The Python Standard Library	180	
<i>Exercise 9-13: Dice</i>	181	
<i>Exercise 9-14: Lottery</i>	181	
<i>Exercise 9-15: Lottery Analysis</i>	181	
<i>Exercise 9-16: Python Module of the Week</i>	181	
Styling Classes	181	
Summary	182	

10	FILES AND EXCEPTIONS	183
Reading from a File	184	
Reading an Entire File	184	
File Paths	185	
Reading Line by Line	187	
Making a List of Lines from a File	188	
Working with a File's Contents	188	

Large Files: One Million Digits	189
Is Your Birthday Contained in Pi?	190
Exercise 10-1: Learning Python	191
Exercise 10-2: Learning C	191
Writing to a File.	191
Writing to an Empty File	191
Writing Multiple Lines	192
Appending to a File.	193
Exercise 10-3: Guest	193
Exercise 10-4: Guest Book	193
Exercise 10-5: Programming Poll.	193
Exceptions.	194
Handling the ZeroDivisionError Exception	194
Using try-except Blocks	194
Using Exceptions to Prevent Crashes	195
The else Block.	196
Handling the FileNotFoundError Exception	197
Analyzing Text	198
Working with Multiple Files	199
Failing Silently	200
Deciding Which Errors to Report.	201
Exercise 10-6: Addition	201
Exercise 10-7: Addition Calculator	202
Exercise 10-8: Cats and Dogs	202
Exercise 10-9: Silent Cats and Dogs	202
Exercise 10-10: Common Words	202
Storing Data	202
Using json.dump() and json.load().	203
Saving and Reading User-Generated Data	204
Refactoring.	206
Exercise 10-11: Favorite Number	208
Exercise 10-12: Favorite Number Remembered	208
Exercise 10-13: Verify User	208
Summary	208
11	
TESTING YOUR CODE	209
Testing a Function	210
Unit Tests and Test Cases	211
A Passing Test	211
A Failing Test	212
Responding to a Failed Test	213
Adding New Tests.	214
Exercise 11-1: City, Country	215
Exercise 11-2: Population.	216
Testing a Class	216
A Variety of Assert Methods	216
A Class to Test	217
Testing the AnonymousSurvey Class	218
The setUp() Method.	220
Exercise 11-3: Employee	221
Summary	222

PART II: PROJECTS

223

PROJECT 1: ALIEN INVASION

12

A SHIP THAT FIRES BULLETS

227

Planning Your Project	228
Installing Pygame.	228
Starting the Game Project	229
Creating a Pygame Window and Responding to User Input	229
Setting the Background Color	230
Creating a Settings Class	231
Adding the Ship Image	232
Creating the Ship Class	233
Drawing the Ship to the Screen.	235
Refactoring: The _check_events() and _update_screen() Methods	236
The _check_events() Method	236
The _update_screen() Method	237
Exercise 12-1: Blue Sky	238
Exercise 12-2: Game Character	238
Piloting the Ship.	238
Responding to a Keypress	238
Allowing Continuous Movement	239
Moving Both Left and Right.	240
Adjusting the Ship's Speed.	241
Limiting the Ship's Range	243
Refactoring _check_events()	243
Pressing Q to Quit.	244
Running the Game in Fullscreen Mode.	244
A Quick Recap	245
alien_invasion.py	245
settings.py	246
ship.py	246
Exercise 12-3: Pygame Documentation	246
Exercise 12-4: Rocket	246
Exercise 12-5: Keys.	246
Shooting Bullets	246
Adding the Bullet Settings.	247
Creating the Bullet Class	247
Storing Bullets in a Group	248
Firing Bullets.	249
Deleting Old Bullets.	250
Limiting the Number of Bullets.	251
Creating the _update_bullets() Method.	252
Exercise 12-6: Sideways Shooter	253
Summary	253

13	
ALIENS!	255
Reviewing the Project	256
Creating the First Alien	256
Creating the Alien Class	257
Creating an Instance of the Alien	258
Building the Alien Fleet	259
Determining How Many Aliens Fit in a Row	260
Creating a Row of Aliens	260
Refactoring <code>_create_fleet()</code>	262
Adding Rows	262
<i>Exercise 13-1: Stars</i>	264
<i>Exercise 13-2: Better Stars</i>	264
Making the Fleet Move	265
Moving the Aliens Right	265
Creating Settings for Fleet Direction	266
Checking Whether an Alien Has Hit the Edge	266
Dropping the Fleet and Changing Direction	267
<i>Exercise 13-3: Raindrops</i>	268
<i>Exercise 13-4: Steady Rain</i>	268
Shooting Aliens	268
Detecting Bullet Collisions	268
Making Larger Bullets for Testing	270
Repopulating the Fleet	270
Speeding Up the Bullets	271
Refactoring <code>_update_bullets()</code>	271
<i>Exercise 13-5: Sideways Shooter Part 2</i>	272
Ending the Game	272
Detecting Alien and Ship Collisions	272
Responding to Alien and Ship Collisions	273
Aliens that Reach the Bottom of the Screen	276
Game Over!	276
Identifying When Parts of the Game Should Run	277
<i>Exercise 13-6: Game Over</i>	278
Summary	278
14	
SCORING	279
Adding the Play Button	280
Creating a Button Class	280
Drawing the Button to the Screen	281
Starting the Game	283
Resetting the Game	283
Deactivating the Play Button	284
Hiding the Mouse Cursor	284
<i>Exercise 14-1: Press P to Play</i>	285
<i>Exercise 14-2: Target Practice</i>	285
Leveling Up	285
Modifying the Speed Settings	285
Resetting the Speed	287

<i>Exercise 14-3: Challenging Target Practice</i>	288
<i>Exercise 14-4: Difficulty Levels</i>	288
Scoring	288
Displaying the Score	288
Making a Scoreboard	289
Updating the Score as Aliens Are Shot Down	291
Resetting the Score	291
Making Sure to Score All Hits	292
Increasing Point Values	292
Rounding the Score	293
High Scores	294
Displaying the Level	296
Displaying the Number of Ships	298
<i>Exercise 14-5: All-Time High Score</i>	301
<i>Exercise 14-6: Refactoring</i>	301
<i>Exercise 14-7: Expanding the Game</i>	302
<i>Exercise 14-8: Sideways Shooter, Final Version</i>	302
Summary	302

PROJECT 2: DATA VISUALIZATION

15	
GENERATING DATA	305
Installing Matplotlib	306
Plotting a Simple Line Graph	306
Changing the Label Type and Line Thickness	307
Correcting the Plot	309
Using Built-in Styles	310
Plotting and Styling Individual Points with <code>scatter()</code>	310
Plotting a Series of Points with <code>scatter()</code>	312
Calculating Data Automatically	312
Defining Custom Colors	314
Using a Colormap	314
Saving Your Plots Automatically	315
<i>Exercise 15-1: Cubes</i>	315
<i>Exercise 15-2: Colored Cubes</i>	315
Random Walks	315
Creating the RandomWalk Class	316
Choosing Directions	316
Plotting the Random Walk	317
Generating Multiple Random Walks	318
Styling the Walk	319
<i>Exercise 15-3: Molecular Motion</i>	323
<i>Exercise 15-4: Modified Random Walks</i>	323
<i>Exercise 15-5: Refactoring</i>	323
Rolling Dice with Plotly	323
Installing Plotly	324
Creating the Die Class	324
Rolling the Die	325

Analyzing the Results	325
Making a Histogram	326
Rolling Two Dice	328
Rolling Dice of Different Sizes	329
<i>Exercise 15-6: Two D8s</i>	331
<i>Exercise 15-7: Three Dice</i>	331
<i>Exercise 15-8: Multiplication</i>	331
<i>Exercise 15-9: Die Comprehensions</i>	331
<i>Exercise 15-10: Practicing with Both Libraries</i>	331
Summary	331

16 DOWNLOADING DATA 333

The CSV File Format	334
Parsing the CSV File Headers	334
Printing the Headers and Their Positions	335
Extracting and Reading Data	336
Plotting Data in a Temperature Chart	336
The datetime Module	337
Plotting Dates	338
Plotting a Longer Timeframe	340
Plotting a Second Data Series	340
Shading an Area in the Chart	342
Error Checking	343
Downloading Your Own Data	345
<i>Exercise 16-1: Sitka Rainfall</i>	346
<i>Exercise 16-2: Sitka–Death Valley Comparison</i>	346
<i>Exercise 16-3: San Francisco</i>	346
<i>Exercise 16-4: Automatic Indexes</i>	347
<i>Exercise 16-5: Explore</i>	347
Mapping Global Data Sets: JSON Format	347
Downloading Earthquake Data	347
Examining JSON Data	347
Making a List of All Earthquakes	350
Extracting Magnitudes	350
Extracting Location Data	351
Building a World Map	351
A Different Way of Specifying Chart Data	353
Customizing Marker Size	353
Customizing Marker Colors	354
Other Colorscales	356
Adding Hover Text	356
<i>Exercise 16-6: Refactoring</i>	357
<i>Exercise 16-7: Automated Title</i>	357
<i>Exercise 16-8: Recent Earthquakes</i>	358
<i>Exercise 16-9: World Fires</i>	358
Summary	358

17 WORKING WITH APIS 359

Using a Web API	359
Git and GitHub	360
Requesting Data Using an API Call	360
Installing Requests	361
Processing an API Response	361
Working with the Response Dictionary	362
Summarizing the Top Repositories	364
Monitoring API Rate Limits	365
Visualizing Repositories Using Plotly	366
Refining Plotly Charts	368
Adding Custom Tooltips	369
Adding Clickable Links to Our Graph	370
More About Plotly and the GitHub API	371
The Hacker News API	372
<i>Exercise 17-1: Other Languages</i>	375
<i>Exercise 17-2: Active Discussions</i>	375
<i>Exercise 17-3: Testing python_repos.py</i>	375
<i>Exercise 17-4: Further Exploration</i>	375
Summary	375

PROJECT 3: WEB APPLICATIONS

18 GETTING STARTED WITH DJANGO 379

Setting Up a Project	380
Writing a Spec	380
Creating a Virtual Environment	380
Activating the Virtual Environment	381
Installing Django	381
Creating a Project in Django	382
Creating the Database	382
Viewing the Project	383
<i>Exercise 18-1: New Projects</i>	384
Starting an App	384
Defining Models	385
Activating Models	386
The Django Admin Site	387
Defining the Entry Model	390
Migrating the Entry Model	391
Registering Entry with the Admin Site	391
The Django Shell	392
<i>Exercise 18-2: Short Entries</i>	394
<i>Exercise 18-3: The Django API</i>	394
<i>Exercise 18-4: Pizzeria</i>	394
Making Pages: The Learning Log Home Page	394
Mapping a URL	395
Writing a View	396

Writing a Template	397
Exercise 18-5: Meal Planner.	398
Exercise 18-6: Pizzeria Home Page.	398
Building Additional Pages	398
Template Inheritance	398
The Topics Page	400
Individual Topic Pages.	403
Exercise 18-7: Template Documentation	406
Exercise 18-8: Pizzeria Pages.	406
Summary	407

19 USER ACCOUNTS 409

Allowing Users to Enter Data	410
Adding New Topics	410
Adding New Entries	414
Editing Entries.	418
Exercise 19-1: Blog.	421
Setting Up User Accounts	421
The users App.	421
The Login Page.	422
Logging Out.	424
The Registration Page	426
Exercise 19-2: Blog Accounts	428
Allowing Users to Own Their Data	428
Restricting Access with @login_required.	429
Connecting Data to Certain Users	430
Restricting Topics Access to Appropriate Users	433
Protecting a User's Topics	434
Protecting the edit_entry Page.	434
Associating New Topics with the Current User	435
Exercise 19-3: Refactoring	436
Exercise 19-4: Protecting new_entry	436
Exercise 19-5: Protected Blog	436
Summary	436

20 STYLING AND DEPLOYING AN APP 437

Styling Learning Log	438
The django-bootstrap4 App	438
Using Bootstrap to Style Learning Log	438
Modifying base.html	439
Styling the Home Page Using a Jumbotron	443
Styling the Login Page	444
Styling the Topics Page	445
Styling the Entries on the Topic Page	446
Exercise 20-1: Other Forms	447
Exercise 20-2: Stylish Blog	447
Deploying Learning Log	448
Making a Heroku Account	448
Installing the Heroku CLI.	448

Installing Required Packages	448
Creating a requirements.txt File.	448
Specifying the Python Runtime	449
Modifying settings.py for Heroku	450
Making a Procfile to Start Processes	450
Using Git to Track the Project's Files	450
Pushing to Heroku	452
Setting Up the Database on Heroku.	454
Refining the Heroku Deployment	454
Securing the Live Project	456
Committing and Pushing Changes.	457
Setting Environment Variables on Heroku.	458
Creating Custom Error Pages	458
Ongoing Development.	461
The SECRET_KEY Setting	461
Deleting a Project on Heroku	461
Exercise 20-3: Live Blog.	462
Exercise 20-4: More 404s	462
Exercise 20-5: Extended Learning Log	462

Summary	463
-------------------	-----

AFTERWORD 465

A INSTALLATION AND TROUBLESHOOTING 467

Python on Windows	467
Finding the Python Interpreter	467
Adding Python to Your Path Variable.	468
Reinstalling Python	469
Python on macOS	469
Installing Homebrew	469
Installing Python	470
Python on Linux	470
Python Keywords and Built-in Functions.	471
Python Keywords	471
Python Built-in Functions	471

B TEXT EDITORS AND IDES 473

Customizing Sublime Text Settings	474
Converting Tabs to Spaces.	474
Setting the Line Length Indicator	474
Indenting and Unindenting Code Blocks.	474
Commenting Out Blocks of Code	475
Saving Your Configuration	475
Further Customizations.	475
Other Text Editors and IDEs.	475
IDLE	475
Geany.	476
Emacs and Vim.	476

Atom	476
Visual Studio Code	476
PyCharm	476
Jupyter Notebooks	477

C

GETTING HELP479

First Steps	479
Try It Again	480
Take a Break	480
Refer to This Book’s Resources	480
Searching Online	481
Stack Overflow	481
The Official Python Documentation	481
Official Library Documentation	482
r/learnpython	482
Blog Posts	482
Internet Relay Chat	482
Making an IRC Account	482
Channels to Join	483
IRC Culture	483
Slack	483
Discord	484

D

USING GIT FOR VERSION CONTROL485

Installing Git	486
Installing Git on Windows	486
Installing Git on macOS	486
Installing Git on Linux	486
Configuring Git	486
Making a Project	486
Ignoring Files	487
Initializing a Repository	487
Checking the Status	487
Adding Files to the Repository	488
Making a Commit	488
Checking the Log	489
The Second Commit	489
Reverting a Change	490
Checking Out Previous Commits	491
Deleting the Repository	493

INDEX

495