

2ND EDITION

WORLDWIDE

BESTSELLER

LEARN PYTHON —

FAST!

OVER 500,000

2ND EDITION

COPIES SOLD

P

PYTHON

YT

Python Crash Course is the world's best-selling guide

- Create and customize web apps and deploy them

CRASH COURSE

H

to the Python programming language. This fast-paced,
safely online

A HANDS-ON, PROJECT-BASED

thorough introduction to programming with Python will

O

- Deal with mistakes and errors so you can solve your

INTRODUCTION TO PROGRAMMING

have you writing programs, solving problems, and
own programming problems
making things that work in no time.

N

This updated second edition has been thoroughly revised

In the first half of the book, you'll learn basic program-

C

to reflect the latest in Python code and practices. The
ming concepts, such as variables, lists, classes, and

ERIC MATTHEWS

first half of the book includes improved coverage of
loops, and practice writing clean code with exercises
topics like f-strings, constants, and managing data. In the

R

for each topic. You'll also learn how to make your
second half, the code for the projects has been updated
programs interactive and test your code safely before
with better structure, cleaner syntax, and more popular

A

adding it to a project. In the second half, you'll put and up-to-date libraries and tools, like Plotly and the your new knowledge into practice with three substantial

S

latest version of Django. (For a full list of updates, see projects: a *Space Invaders*–inspired arcade game, a the Preface.)

H

set of data visualizations with Python's handy libraries, and a simple web app you can deploy online.

If you've been thinking about digging into programming,

C

Python Crash Course will get you writing real programs

As you work through the book, you'll learn how to:

fast. Why wait any longer? Start your engines and code!

OU

- Use powerful Python libraries and tools, including

ABOUT THE AUTHOR

Pygame, Matplotlib, Plotly, and Django

Eric Matthes is a high school science, math, and program-

- Make 2D games that respond to keypresses and mouse clicks, and that increase in difficulty
- programs since he was five years old and is the author

R

mouse clicks, and that increase in difficulty

programs since he was five years old and is the author

SE

- Use data to generate interactive visualizations of the *Python Flash Cards*, also from No Starch Press.

C O V E R S P Y T H O N 3 . X

MATTHES

THE FINEST IN GEEK ENTERTAINMENT™

www.nostarch.com

\$39.95 (\$53.95 CDN)

P P S

Y R H

T O E

H O N G R L V E

A I

M N

MI :

NG L

“I L I E FLAT.”

ANG

FSC FPO

This book uses a durable binding that won't snap shut

UAGES/

PRAISE FOR *Python Crash Course*

“It has been interesting to see No Starch Press producing future classics that should be alongside the more traditional programming books. *Python Crash Course* is one of those books.”

—[GreG Laden, ScienceBLoGS](#)

“Deals with some rather complex projects and lays them out in a consistent, logical, and pleasant manner that draws the reader into the subject.”

—[FuLL circLe MaGazine](#)

“Well presented with good explanations of the code snippets. The book works with you, one small step at a time, building more complex code, explaining what's going on all the way.”

—FLickThrouGh reviewS

“Learning Python with *Python Crash Course* was an extremely positive experience! A great choice if you’re new to Python.”

—Mikke GoeS codinG

“Does what it says on the tin, and does it really well. . . . Presents a large number of useful exercises as well as three challenging and entertaining projects.”

—reaLPyThon.coM

“A fast-paced but comprehensive introduction to programming with Python, *Python Crash Course* is another superb book to add to your library and help you finally master Python.”

—TuToriaLedGe.neT

“A brilliant option for complete beginners without any coding experience. If you’re looking for a solid, uncomplicated intro to this very deep language, I have to recommend this book.”

—whaTPixeL.coM

“Contains literally everything you need to know about Python and even more.”

—FireBearSTudio.coM

P y t h o n

C r a s h C o u r s e

2nd edition

a hands-on, Project-Based

introduction to Programming

by Eric Matthes

San Francisco

Python Crash Course, 2nd edition. Copyright © 2019 by Eric Matthes.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means,

electronic or mechanical, including photocopying, recording, or by any information storage or retrieval

system, without the prior written permission of the copyright owner and the publisher.

ISBN-10: 1-59327-928-0

ISBN-13: 978-1-59327-928-8

Publisher: William Pollock

Production Editor: Riley Hoffman

Cover Illustration: Josh Ellingson

Cover and Interior Design: Octopod Studios

Developmental Editor: Liz Chadwick

Technical Reviewer: Kenneth Love

Copyeditor: Anne Marie Walker

Compositors: Riley Hoffman and Happenstance Type-O-Rama

Proofreader: James Fraleigh

For information on distribution, translations, or bulk sales, please contact
No Starch Press, Inc. directly:

No Starch Press, Inc.

245 8th Street, San Francisco, CA 94103

phone: 1.415.863.9900; info@nostarch.com

www.nostarch.com

The Library of Congress has catalogued the first edition as follows:

Matthes, Eric, 1972-

Python crash course : a hands-on, project-based introduction to
programming / by Eric Matthes.

pages cm

Includes index.

Summary: "A project-based introduction to programming in Python, with
exercises. Covers general

programming concepts, Python fundamentals, and problem solving.
Includes three projects - how to

create a simple video game, use data visualization techniques to make
graphs and charts, and build

an interactive web application"-- Provided by publisher.

ISBN 978-1-59327-603-4 -- ISBN 1-59327-603-6

1. Python (Computer program language) I. Title.

QA76.73.P98M38 2015

005.13'3--dc23

2015018135

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other

product and company names mentioned herein may be the trademarks of their respective owners. Rather

than use a trademark symbol with every occurrence of a trademarked name, we are using the names only

in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the

trademark.

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution

has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any

liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or

indirectly by the information contained in it.

About the Author

Eric Matthes is a high school science and math teacher living in Alaska, where he teaches an introductory Python course. He has been writing

programs since he was five years old. Eric currently focuses on writing software that addresses inefficiencies in education and brings the benefits of open source software to the field of education. In his spare time he enjoys climbing mountains and spending time with his family.

About the Technical Reviewer

Kenneth Love has been a Python programmer, teacher, and conference organizer for many years. He has spoken and taught at many conferences, been a Python and Django freelancer, and is currently a software engineer for O'Reilly Media. Kenneth is co-creator of the django-braces package, which provides several handy mixins for Django's class-based views. You can keep up with him on Twitter at @kennethlove.

For my father, who always made time to
answer my questions about programming,
and for Ever, who is just beginning to ask me
his questions

B r i e f C o n t e n t s

[Preface to the Second Edition xxvii](#)

[Acknowledgments xxxi](#)

Introduction	
.....	xxxiii

Part I: Basics 1

Chapter 1: Getting Started	
.....	3

Chapter 2: Variables and Simple Data Types	
.....	15

Chapter 3: Introducing Lists	
.....	33

Chapter 4: Working with Lists	
.....	49

Chapter 5: if Statements	
.....	71

Chapter 6: Dictionaries	
.....	91

Chapter 7: User Input and while Loops	
.....	113

Chapter 8: Functions	
.....	129

Chapter 9: Classes	
.....	157

Chapter 10: Files and Exceptions	
.....	183

Chapter 11: Testing Your Code	
.....	209

Part II: Projects 223

Project 1: Alien Invasion

Chapter 12: A Ship that Fires Bullets 227

Chapter 13: Aliens! 255

Chapter 14: Scoring 279

Project 2: Data Visualization

Chapter 15: Generating Data 305

Chapter 16: Downloading Data 333

Chapter 17: Working with APIs 359

Project 3: Web Applications

Chapter 18: Getting Started with Django 379

Chapter 19: User Accounts 409

Chapter 20: Styling and Deploying an App 437

Afterword 465

Appendix A: Installation and Troubleshooting 467

[Appendix B: Text Editors and IDEs 473](#)

[Appendix C: Getting Help 479](#)

[Appendix D: Using Git for Version Control 485](#)

[Index 495](#)

x Brief Contents

C o n t e n t s i n D e t a i l

[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: BaSicS

1

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

<i>Exercise 2-1: Simple Message</i>	19
<i>Exercise 2-2: Simple Messages</i>	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
<i>Exercise 2-3: Personal Message</i>	25
<i>Exercise 2-4: Name Cases</i>	25
<i>Exercise 2-5: Famous Quote</i>	25
<i>Exercise 2-6: Famous Quote 2</i>	25
<i>Exercise 2-7: Stripping Names</i>	25
Numbers	25
Integers	26
Floats	26

<u>Integers and Floats</u>	<u>27</u>
<u>Underscores in Numbers</u>	<u>28</u>
<u>Multiple Assignment</u>	<u>28</u>
<u>Constants</u>	<u>28</u>
<u>Exercise 2-8: Number Eight</u>	<u>29</u>
<u>Exercise 2-9: Favorite Number</u>	<u>29</u>
<u>Comments</u>	<u>29</u>
<u>How Do You Write Comments?</u>	<u>29</u>
<u>What Kind of Comments Should You Write?</u>	<u>29</u>
<u>Exercise 2-10: Adding Comments</u>	<u>30</u>
<u>The Zen of Python</u>	<u>30</u>
<u>Exercise 2-11: Zen of Python</u>	<u>31</u>
<u>Summary</u>	<u>32</u>

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

<i>Exercise 3-7: Shrinking Guest List</i>	
<u>43</u>	
<i>Organizing a List</i>	
<u>... 43</u>	
<i>Sorting a List Permanently with the sort() Method</i>	
<u>. 43</u>	
<i>Sorting a List Temporarily with the sorted() Function</i>	
<u>.. 44</u>	
<i>Printing a List in Reverse Order</i>	
<u>45</u>	
<i>Finding the Length of a List</i>	
<u>45</u>	
<i>Exercise 3-8: Seeing the World</i>	46
<i>Exercise 3-9: Dinner Guests</i>	
<u>46</u>	
<i>Exercise 3-10: Every Function</i>	
<u>46</u>	
xii Contents in Detail	
<i>Avoiding Index Errors When Working with Lists</i>	
<u>..... 46</u>	
<i>Exercise 3-11: Intentional Error</i>	
<u>48</u>	
<i>Summary</i>	
<u>.. 48</u>	

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

<u>Using the range() Function</u>	<u>57</u>
<u>Using range() to Make a List of Numbers</u>	<u>58</u>
<u>Simple Statistics with a List of Numbers</u>	<u>59</u>
<u>List Comprehensions</u>	<u>59</u>
<u>Exercise 4-3: Counting to Twenty</u>	<u>60</u>
<u>Exercise 4-4: One Million</u>	<u>60</u>
<u>Exercise 4-5: Summing a Million</u>	<u>60</u>
<u>Exercise 4-6: Odd Numbers</u>	<u>60</u>
<u>Exercise 4-7: Threes</u>	<u>60</u>
<u>Exercise 4-8: Cubes</u>	<u>60</u>
<u>Exercise 4-9: Cube Comprehension</u>	<u>60</u>
<u>Working with Part of a List</u>	<u>61</u>
<u>Slicing a List</u>	<u>61</u>
<u>Looping Through a Slice</u>	<u>62</u>
<u>Copying a List</u>	<u>63</u>
<u>Exercise 4-10: Slices</u>	<u>65</u>

<i>Exercise 4-11: My Pizzas, Your Pizzas</i>	65
<i>Exercise 4-12: More Loops</i>	65
Tuples	65
Defining a Tuple	66
Looping Through All Values in a Tuple	67
Writing over a Tuple	67
<i>Exercise 4-13: Buffet</i>	68
Styling Your Code	68
The Style Guide	68
Indentation	69
Line Length	69
Blank Lines	69
Other Style Guidelines	70
<i>Exercise 4-14: PEP 8</i>	70
<i>Exercise 4-15: Code Review</i>	70
Summary	70

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

Exercise 5-1: Conditional Tests 78

Exercise 5-2: More Conditional Tests 78

<u>if Statements</u>	<u>78</u>
<u>Simple if Statements</u>	<u>78</u>
<u>if-else Statements</u>	<u>79</u>
<u>The if-elif-else Chain</u>	<u>80</u>
<u>Using Multiple elif Blocks</u>	<u>82</u>
<u>Omitting the else Block</u>	<u>82</u>
<u>Testing Multiple Conditions</u>	<u>83</u>
<u>Exercise 5-3: Alien Colors #1</u>	<u>84</u>
<u>Exercise 5-4: Alien Colors #2</u>	<u>84</u>
<u>Exercise 5-5: Alien Colors #3</u>	<u>85</u>
<u>Exercise 5-6: Stages of Life</u>	<u>85</u>
<u>Exercise 5-7: Favorite Fruit</u>	<u>85</u>
<u>Using if Statements with Lists</u>	<u>85</u>
<u>Checking for Special Items</u>	<u>86</u>
<u>Checking That a List Is Not Empty</u>	<u>87</u>

<u>Using Multiple Lists</u>	<u>88</u>
<u><i>Exercise 5-8: Hello Admin</i></u>	<u>89</u>
<u><i>Exercise 5-9: No Users</i></u>	<u>89</u>
<u><i>Exercise 5-10: Checking Usernames</i></u>	<u>89</u>
<u><i>Exercise 5-11: Ordinal Numbers</i></u>	<u>89</u>
<u>Styling Your if Statements</u>	<u>90</u>
<u><i>Exercise 5-12: Styling if statements</i></u>	<u>90</u>
<u><i>Exercise 5-13: Your Ideas</i></u>	<u>90</u>
<u>Summary</u>	<u>90</u>

6

dictionaryS 91

<u>A Simple Dictionary</u>	<u>92</u>
<u>Working with Dictionaries</u>	<u>92</u>
<u>Accessing Values in a Dictionary</u>	<u>93</u>
<u>Adding New Key-Value Pairs</u>	<u>93</u>
<u>Starting with an Empty Dictionary</u>	<u>94</u>

Modifying Values in a Dictionary
95

Removing Key-Value Pairs
96

A Dictionary of Similar Objects
97

Using .get() to Access Values
98

xiv Contents in Detail

Exercise 6-1: Person 99

Exercise 6-2: Favorite Numbers
99

Exercise 6-3: Glossary 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

Exercise 6-4: Glossary 2 105

Exercise 6-5: Rivers 105

Exercise 6-6: Polling 105

Nesting
. . 106

A List of Dictionaries 106

A List in a Dictionary 108

A Dictionary in a Dictionary
110

Exercise 6-7: People 112

Exercise 6-8: Pets 112

Exercise 6-9: Favorite Places
112

Exercise 6-10: Favorite Numbers
112

Exercise 6-11: Cities 112

Exercise 6-12: Extensions 112

Summary
. 112

7

uSer inPut and while looPS

113

How the input() Function Works
. . . . 114

Writing Clear Prompts 114

<u>Using <code>int()</code> to Accept Numerical Input</u>	<u>115</u>
<u>The Modulo Operator</u>	<u>116</u>
<u><i>Exercise 7-1: Rental Car</i></u>	<u>117</u>
<u><i>Exercise 7-2: Restaurant Seating</i></u>	<u>117</u>
<u><i>Exercise 7-3: Multiples of Ten</i></u>	<u>117</u>
<u>Introducing <code>while</code> Loops</u>	<u>118</u>
<u>The <code>while</code> Loop in Action</u>	<u>118</u>
<u>Letting the User Choose When to Quit</u>	<u>118</u>
<u>Using a Flag</u>	<u>120</u>
<u>Using <code>break</code> to Exit a Loop</u>	<u>121</u>
<u>Using <code>continue</code> in a Loop</u>	<u>122</u>
<u>Avoiding Infinite Loops</u>	<u>122</u>
<u><i>Exercise 7-4: Pizza Toppings</i></u>	<u>123</u>
<u><i>Exercise 7-5: Movie Tickets</i></u>	<u>123</u>
<u><i>Exercise 7-6: Three Exits</i></u>	<u>124</u>
<u><i>Exercise 7-7: Infinity</i></u>	<u>124</u>

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

Exercise 7-8: Deli 127

Exercise 7-9: No Pastrami 127

Exercise 7-10: Dream Vacation. 127

Summary 127

Contents in Detail xv

8

functionS 129

Defining a Function 130

Passing Information to a Function 130

Arguments and Parameters 131

Exercise 8-1: Message. 131

<i>Exercise 8-2: Favorite Book</i>	
<u>131</u>	
Passing Arguments	
... 131	
Positional Arguments	132
Keyword Arguments	133
Default Values	134
Equivalent Function Calls	
<u>135</u>	
Avoiding Argument Errors	136
<i>Exercise 8-3: T-Shirt</i>	
<u>137</u>	
<i>Exercise 8-4: Large Shirts</i>	
<u>137</u>	
<i>Exercise 8-5: Cities</i>	137
Return Values	
... 137	
Returning a Simple Value	
<u>138</u>	
Making an Argument Optional	
<u>138</u>	
Returning a Dictionary	140
Using a Function with a while Loop	
<u>141</u>	

<i><u>Exercise 8-6: City Names</u></i>	<i><u>142</u></i>
<i><u>Exercise 8-7: Album</u></i>	<i><u>142</u></i>
<i><u>Exercise 8-8: User Albums</u></i>	<i><u>142</u></i>
<u>Passing a List</u>	<u>143</u>
<u>Modifying a List in a Function</u>	<u>143</u>
<u>Preventing a Function from Modifying a List</u>	<u>145</u>
<i><u>Exercise 8-9: Messages</u></i>	<i><u>146</u></i>
<i><u>Exercise 8-10: Sending Messages</u></i>	<i><u>146</u></i>
<i><u>Exercise 8-11: Archived Messages</u></i>	<i><u>146</u></i>
<u>Passing an Arbitrary Number of Arguments</u>	<u>147</u>
<u>Mixing Positional and Arbitrary Arguments</u>	<u>148</u>
<u>Using Arbitrary Keyword Arguments</u>	<u>148</u>
<i><u>Exercise 8-12: Sandwiches</u></i>	<i><u>150</u></i>
<i><u>Exercise 8-13: User Profile</u></i>	<i><u>150</u></i>
<i><u>Exercise 8-14: Cars</u></i>	<i><u>150</u></i>
<u>Storing Your Functions in Modules</u>	<u>150</u>

<u>Importing an Entire Module</u>	<u>150</u>
<u>Importing Specific Functions</u>	<u>152</u>
<u>Using as to Give a Function an Alias</u>	<u>152</u>
<u>Using as to Give a Module an Alias</u>	<u>153</u>
<u>Importing All Functions in a Module</u>	<u>153</u>
<u>Styling Functions</u>	<u>154</u>
<u>Exercise 8-15: Printing Models</u>	<u>155</u>
<u>Exercise 8-16: Imports</u>	<u>155</u>
<u>Exercise 8-17: Styling Functions</u>	<u>155</u>
<u>Summary</u>	<u>155</u>

xvi Contents in Detail

2

claSSeS 157

<u>Creating and Using a Class</u>	<u>158</u>
---	------------

<u>Creating the Dog Class</u>	<u>158</u>
---	------------

<u>Making an Instance from a Class</u>	<u>160</u>
<u><i>Exercise 9-1: Restaurant.</i></u>	<u>162</u>
<u><i>Exercise 9-2: Three Restaurants.</i></u>	<u>162</u>
<u><i>Exercise 9-3: Users</i></u>	<u>162</u>
<u>Working with Classes and Instances</u>	<u>162</u>
<u>The Car Class</u>	<u>162</u>
<u>Setting a Default Value for an Attribute</u>	<u>163</u>
<u>Modifying Attribute Values</u>	<u>164</u>
<u><i>Exercise 9-4: Number Served</i></u>	<u>167</u>
<u><i>Exercise 9-5: Login Attempts</i></u>	<u>167</u>
<u>Inheritance</u>	<u>167</u>
<u>The <code>__init__()</code> Method for a Child Class</u>	<u>167</u>
<u>Defining Attributes and Methods for the Child Class</u>	<u>169</u>
<u>Overriding Methods from the Parent Class</u>	<u>170</u>
<u>Instances as Attributes</u>	<u>170</u>

<u>Modeling Real-World Objects</u>	<u>173</u>
<u>Exercise 9-6: Ice Cream Stand</u>	<u>173</u>
<u>Exercise 9-7: Admin</u>	<u>173</u>
<u>Exercise 9-8: Privileges</u>	<u>173</u>
<u>Exercise 9-9: Battery Upgrade</u>	<u>174</u>
<u>Importing Classes</u>	<u>174</u>
<u>Importing a Single Class</u>	<u>174</u>
<u>Storing Multiple Classes in a Module</u>	<u>175</u>
<u>Importing Multiple Classes from a Module</u>	<u>177</u>
<u>Importing an Entire Module</u>	<u>177</u>
<u>Importing All Classes from a Module</u>	<u>177</u>
<u>Importing a Module into a Module</u>	<u>178</u>
<u>Using Aliases</u>	<u>179</u>
<u>Finding Your Own Workflow</u>	<u>179</u>
<u>Exercise 9-10: Imported Restaurant</u>	<u>180</u>

<i><u>Exercise 9-11: Imported Admin</u></i>	
<u>180</u>	
<i><u>Exercise 9-12: Multiple Modules</u></i>	
<u>180</u>	
<u>The Python Standard Library</u>	
<u>.... 180</u>	
<i><u>Exercise 9-13: Dice</u></i>	<i>181</i>
<i><u>Exercise 9-14: Lottery</u></i>	<i>181</i>
<i><u>Exercise 9-15: Lottery Analysis</u></i>	
<u>181</u>	
<i><u>Exercise 9-16: Python Module of the Week</u></i>	
<u>181</u>	
<u>Styling Classes</u>	
<u>... 181</u>	
<u>Summary</u>	
<u>. 182</u>	
<u>10</u>	
<u>fileS and excePtionS</u>	
<u>183</u>	
<u>Reading from a File</u>	
<u>.. 184</u>	
<u>Reading an Entire File</u>	184
<u>File Paths</u>	185
<u>Reading Line by Line</u>	187

<u>Making a List of Lines from a File</u>	
<u>188</u>	

<u>Working with a File's Contents</u>	
<u>188</u>	

Contents in Detail **xvii**

<u>Large Files: One Million Digits</u>	
<u>189</u>	

<u>Is Your Birthday Contained in Pi?</u>	
<u>190</u>	

<u><i>Exercise 10-1: Learning Python</i></u>	
<u>191</u>	

<u><i>Exercise 10-2: Learning C</i></u>	<u>191</u>
---	------------

<u>Writing to a File</u>	
<u>... 191</u>	

<u>Writing to an Empty File</u>	<u>191</u>
---------------------------------------	------------

<u>Writing Multiple Lines</u>	<u>192</u>
-------------------------------------	------------

<u>Appending to a File</u>	<u>193</u>
----------------------------------	------------

<u><i>Exercise 10-3: Guest</i></u>	<u>193</u>
--	------------

<u><i>Exercise 10-4: Guest Book</i></u>	<u>193</u>
---	------------

<u><i>Exercise 10-5: Programming Poll</i></u>	
<u>193</u>	

<u>Exceptions</u>	
<u>.. 194</u>	

<u>Handling the ZeroDivisionError Exception</u>	
<u>194</u>	
<u>Using try-except Blocks</u>	<u>194</u>
<u>Using Exceptions to Prevent Crashes</u>	
<u>195</u>	
<u>The else Block</u>	<u>196</u>
<u>Handling the FileNotFoundError Exception</u>	
<u>197</u>	
<u>Analyzing Text</u>	<u>198</u>
<u>Working with Multiple Files</u>	
<u>199</u>	
<u>Failing Silently</u>	<u>200</u>
<u>Deciding Which Errors to Report</u>	
<u>201</u>	
<u>Exercise 10-6: Addition</u>	<u>201</u>
<u>Exercise 10-7: Addition Calculator</u>	
<u>202</u>	
<u>Exercise 10-8: Cats and Dogs</u>	<u>202</u>
<u>Exercise 10-9: Silent Cats and Dogs</u>	
<u>202</u>	
<u>Exercise 10-10: Common Words</u>	<u>202</u>
<u>Storing Data</u>	
<u>202</u>	

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

<i>Exercise 11-1: City, Country</i>	
<u>215</u>	
<i>Exercise 11-2: Population</i>	216
Testing a Class	
<u>.. 216</u>	
A Variety of Assert Methods	
<u>216</u>	
A Class to Test	217
Testing the AnonymousSurvey Class	
<u>218</u>	
The setUp() Method	220
<i>Exercise 11-3: Employee</i>	221
Summary	
<u>. 222</u>	

xviii Contents in Detail

Part ii: ProjectS

223

Project 1: alien inVaSion

12

a ShiP that fireS BulletS

227

Planning Your Project	
<u>... 228</u>	

<u>Installing Pygame</u>	<u>228</u>
<u>Starting the Game Project</u>	<u>229</u>
<u>Creating a Pygame Window and Responding to User Input</u>	<u>229</u>
<u>Setting the Background Color</u>	<u>230</u>
<u>Creating a Settings Class</u>	<u>231</u>
<u>Adding the Ship Image</u>	<u>232</u>
<u>Creating the Ship Class</u>	<u>233</u>
<u>Drawing the Ship to the Screen</u>	<u>235</u>
<u>Refactoring: The <code>_check_events()</code> and <code>_update_screen()</code> Methods</u>	<u>236</u>
<u>The <code>_check_events()</code> Method</u>	<u>236</u>
<u>The <code>_update_screen()</code> Method</u>	<u>237</u>
<u><i>Exercise 12-1: Blue Sky</i></u>	<u>238</u>
<u><i>Exercise 12-2: Game Character</i></u>	<u>238</u>
<u>Piloting the Ship</u>	<u>238</u>

<u>Responding to a Keypress</u>	<u>238</u>
<u>Allowing Continuous Movement</u>	<u>239</u>
<u>Moving Both Left and Right</u>	<u>240</u>
<u>Adjusting the Ship's Speed</u>	<u>241</u>
<u>Limiting the Ship's Range</u>	<u>243</u>
<u>Refactoring <code>_check_events()</code></u>	<u>243</u>
<u>Pressing Q to Quit</u>	<u>244</u>
<u>Running the Game in Fullscreen Mode</u>	<u>244</u>
<u>A Quick Recap</u>	<u>245</u>
<u><code>alien_invasion.py</code></u>	<u>245</u>
<u><code>settings.py</code></u>	<u>246</u>
<u><code>ship.py</code></u>	<u>246</u>
<u><i>Exercise 12-3: Pygame Documentation</i></u>	<u>246</u>
<u><i>Exercise 12-4: Rocket</i></u>	<u>246</u>
<u><i>Exercise 12-5: Keys</i></u>	<u>246</u>

<u>Shooting Bullets . . .</u>	<u>246</u>
<u>Adding the Bullet Settings</u>	<u>247</u>
<u>Creating the Bullet Class</u>	<u>247</u>
<u>Storing Bullets in a Group</u>	<u>248</u>
<u>Firing Bullets</u>	<u>249</u>
<u>Deleting Old Bullets</u>	<u>250</u>
<u>Limiting the Number of Bullets</u>	<u>251</u>
<u>Creating the _update_bullets() Method</u>	<u>252</u>
<u><i>Exercise 12-6: Sideways Shooter</i></u>	<u>253</u>
<u>Summary</u>	<u>253</u>
Contents in Detail xix	
<u>13</u>	
<u>Aliens! 255</u>	
<u>Reviewing the Project</u>	<u>256</u>
<u>Creating the First Alien</u>	<u>256</u>
<u>Creating the Alien Class</u>	<u>257</u>

<u>Creating an Instance of the Alien</u>	<u>258</u>
<u>Building the Alien Fleet</u>	<u>259</u>
<u>Determining How Many Aliens Fit in a Row</u>	<u>260</u>
<u>Creating a Row of Aliens</u>	<u>260</u>
<u>Refactoring <code>_create_fleet()</code></u>	<u>262</u>
<u>Adding Rows</u>	<u>262</u>
<u><i>Exercise 13-1: Stars</i></u>	<u>264</u>
<u><i>Exercise 13-2: Better Stars</i></u>	<u>264</u>
<u>Making the Fleet Move</u>	<u>265</u>
<u>Moving the Aliens Right</u>	<u>265</u>
<u>Creating Settings for Fleet Direction</u>	<u>266</u>
<u>Checking Whether an Alien Has Hit the Edge</u>	<u>266</u>
<u>Dropping the Fleet and Changing Direction</u>	<u>267</u>
<u><i>Exercise 13-3: Raindrops</i></u>	<u>268</u>
<u><i>Exercise 13-4: Steady Rain</i></u>	<u>268</u>

<u>Shooting Aliens ..</u>	<u>268</u>
<u>Detecting Bullet Collisions</u>	<u>268</u>
<u>Making Larger Bullets for Testing</u>	<u>270</u>
<u>Repopulating the Fleet</u>	<u>270</u>
<u>Speeding Up the Bullets</u>	<u>271</u>
<u>Refactoring _update_bullets()</u>	<u>271</u>
<u>Exercise 13-5: Sideways Shooter Part 2</u>	<u>272</u>
<u>Ending the Game</u>	<u>272</u>
<u>Detecting Alien and Ship Collisions</u>	<u>272</u>
<u>Responding to Alien and Ship Collisions</u>	<u>273</u>
<u>Aliens that Reach the Bottom of the Screen</u>	<u>276</u>
<u>Game Over!</u>	<u>276</u>
<u>Identifying When Parts of the Game Should Run</u>	<u>277</u>
<u>Exercise 13-6: Game Over</u>	<u>278</u>

<u>Summary</u>	<u>278</u>
----------------	------------

14

scoring 279

<u>Adding the Play Button</u>	<u>280</u>
-------------------------------	------------

<u>Creating a Button Class</u>	<u>280</u>
--------------------------------	------------

<u>Drawing the Button to the Screen</u>	<u>281</u>
---	------------

<u>Starting the Game</u>	<u>283</u>
--------------------------	------------

<u>Resetting the Game</u>	<u>283</u>
---------------------------	------------

<u>Deactivating the Play Button</u>	<u>284</u>
-------------------------------------	------------

<u>Hiding the Mouse Cursor</u>	<u>284</u>
--------------------------------	------------

<u>Exercise 14-1: Press P to Play</u>	<u>285</u>
---------------------------------------	------------

<u>Exercise 14-2: Target Practice</u>	<u>285</u>
---------------------------------------	------------

<u>Leveling Up</u>	<u>285</u>
--------------------	------------

<u>Modifying the Speed Settings</u>	<u>285</u>
-------------------------------------	------------

<u>Resetting the Speed</u>	<u>287</u>
----------------------------	------------

xx Contents in Detail

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

Exercise 14-7: Expanding the Game
302

Exercise 14-8: Sideways Shooter, Final Version
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 scatter()
. 310

Plotting a Series of Points with scatter()
312

Calculating Data Automatically
312

<u>Defining Custom Colors</u>	
<u>314</u>	
<u>Using a Colormap</u>	<u>314</u>
<u>Saving Your Plots Automatically</u>	
<u>315</u>	
<u>Exercise 15-1: Cubes</u>	<u>315</u>
<u>Exercise 15-2: Colored Cubes</u>	<u>315</u>
<u>Random Walks</u>	
<u>.. 315</u>	
<u>Creating the RandomWalk() Class</u>	
<u>316</u>	
<u>Choosing Directions</u>	<u>316</u>
<u>Plotting the Random Walk</u>	<u>317</u>
<u>Generating Multiple Random Walks</u>	
<u>318</u>	
<u>Styling the Walk</u>	<u>319</u>
<u>Exercise 15-3: Molecular Motion</u>	
<u>323</u>	
<u>Exercise 15-4: Modified Random Walks</u>	
<u>323</u>	
<u>Exercise 15-5: Refactoring</u>	
<u>323</u>	
<u>Rolling Dice with Plotly</u>	
<u>.... 323</u>	

Installing Plotly 324

Creating the Die Class 324

Rolling the Die 325

Contents in Detail **xxi**

Analyzing the Results
325

Making a Histogram 326

Rolling Two Dice 328

Rolling Dice of Different Sizes
329

Exercise 15-6: Two D8s 331

Exercise 15-7: Three Dice 331

Exercise 15-8: Multiplication
331

Exercise 15-9: Die Comprehensions
331

Exercise 15-10: Practicing with Both Libraries
331

Summary
331

16

downloading data

333

<u>The CSV File Format</u>	<u>334</u>
<u>Parsing the CSV File Headers</u>	<u>334</u>
<u>Printing the Headers and Their Positions</u>	<u>335</u>
<u>Extracting and Reading Data</u>	<u>336</u>
<u>Plotting Data in a Temperature Chart</u>	<u>336</u>
<u>The datetime Module</u>	<u>337</u>
<u>Plotting Dates</u>	<u>338</u>
<u>Plotting a Longer Timeframe</u>	<u>340</u>
<u>Plotting a Second Data Series</u>	<u>340</u>
<u>Shading an Area in the Chart</u>	<u>342</u>
<u>Error Checking</u>	<u>343</u>
<u>Downloading Your Own Data</u>	<u>345</u>
<u><i>Exercise 16-1: Sitka Rainfall</i></u>	<u><i>346</i></u>
<u><i>Exercise 16-2: Sitka–Death Valley Comparison</i></u>	<u><i>346</i></u>
<u><i>Exercise 16-3: San Francisco</i></u>	<u><i>346</i></u>