

brief contents

- | | | |
|----|---|-----|
| | Getting started: Introduction and installation guide | 1 |
| 1 | ■ How to write and test a Python program | 15 |
| 2 | ■ The crow's nest: Working with strings | 35 |
| 3 | ■ Going on a picnic: Working with lists | 55 |
| 4 | ■ Jump the Five: Working with dictionaries | 76 |
| 5 | ■ Howler: Working with files and STDOUT | 92 |
| 6 | ■ Words count: Reading files and STDIN, iterating lists, formatting strings | 107 |
| 7 | ■ Gashlycrumb: Looking items up in a dictionary | 118 |
| 8 | ■ Apples and Bananas: Find and replace | 128 |
| 9 | ■ Dial-a-Curse: Generating random insults from lists of words | 150 |
| 10 | ■ Telephone: Randomly mutating strings | 165 |
| 11 | ■ Bottles of Beer Song: Writing and testing functions | 178 |
| 12 | ■ Ransom: Randomly capitalizing text | 195 |
| 13 | ■ Twelve Days of Christmas: Algorithm design | 207 |

1 How to write and test a Python program 15

- 1.1 Creating your first program 15
- 1.2 Comment lines 16
- 1.3 Testing your program 17
- 1.4 Adding the #! (shebang) line 18
- 1.5 Making a program executable 20
- 1.6 Understanding \$PATH 20
 - Altering your \$PATH* 21
- 1.7 Adding a parameter and help 22
- 1.8 Making the argument optional 24
- 1.9 Running our tests 26
- 1.10 Adding the main() function 26
- 1.11 Adding the get_args() function 27
 - Checking style and errors* 28
- 1.12 Testing hello.py 29
- 1.13 Starting a new program with new.py 30
- 1.14 Using template.py as an alternative to new.py 33

2 The crow's nest: Working with strings 35

- 2.1 Getting started 36
 - How to use the tests* 36 ▪ *Creating programs with new.py* 37
 - Write, test, repeat* 38 ▪ *Defining your arguments* 39
 - Concatenating strings* 41 ▪ *Variable types* 42 ▪ *Getting just part of a string* 43 ▪ *Finding help in the REPL* 44
 - String methods* 44 ▪ *String comparisons* 45 ▪ *Conditional branching* 47 ▪ *String formatting* 48 ▪ *Time to write* 49
- 2.2 Solution 49
- 2.3 Discussion 50
 - Defining the arguments with get_args()* 50 ▪ *The main() thing* 51 ▪ *Classifying the first character of a word* 51
 - Printing the results* 52 ▪ *Running the test suite* 52
- 2.4 Going further 53

3 Going on a picnic: Working with lists 55

- 3.1 Starting the program 56
- 3.2 Writing picnic.py 58

9.2	Solution	157
9.3	Discussion	159
	<i>Defining the arguments</i>	159
	<i>Using parser.error()</i>	160
	<i>Program exit values and STDERR</i>	160
	<i>Controlling randomness with random.seed()</i>	161
	<i>Iterating with range() and using throwaway variables</i>	162
	<i>Constructing the insults</i>	162
9.4	Going further	163

10 Telephone: Randomly mutating strings 165

10.1	Writing telephone.py	167
	<i>Calculating the number of mutations</i>	168
	<i>The mutation space</i>	169
	<i>Selecting the characters to mutate</i>	169
	<i>Mutating a string</i>	172
	<i>Time to write</i>	173
10.2	Solution	173
10.3	Discussion	175
	<i>Mutating a string</i>	175
	<i>Using a list instead of a str</i>	176
10.4	Going further	177

11 Bottles of Beer Song: Writing and testing functions 178

11.1	Writing bottles.py	179
	<i>Counting down</i>	180
	<i>Writing a function</i>	181
	<i>Writing a test for verse()</i>	182
	<i>Using the verse() function</i>	186
11.2	Solution	187
11.3	Discussion	189
	<i>Counting down</i>	189
	<i>Test-driven development</i>	189
	<i>The verse() function</i>	190
	<i>Iterating through the verses</i>	191
	<i>1,500 other solutions</i>	194
11.4	Going further	194

12 Ransom: Randomly capitalizing text 195

12.1	Writing ransom.py	197
	<i>Mutating the text</i>	197
	<i>Flipping a coin</i>	198
	<i>Creating a new string</i>	198
12.2	Solution	199
12.3	Discussion	200
	<i>Iterating through elements in a sequence</i>	200
	<i>Writing a function to choose the letter</i>	202
	<i>Another way to write list.append()</i>	202
	<i>Using a str instead of a list</i>	203
	<i>Using a list comprehension</i>	203
	<i>Using a map() function</i>	204