

Python for Everybody

Exploring Data Using Python 3

Charles R. Severance

Contents

| | | |
|----------|---|-----------|
| 1 | Why should you learn to write programs? | 1 |
| 1.1 | Creativity and motivation | 2 |
| 1.2 | Computer hardware architecture | 3 |
| 1.3 | Understanding programming | 4 |
| 1.4 | Words and sentences | 5 |
| 1.5 | Conversing with Python | 6 |
| 1.6 | Terminology: interpreter and compiler | 8 |
| 1.7 | Writing a program | 10 |
| 1.8 | What is a program? | 10 |
| 1.9 | The building blocks of programs | 11 |
| 1.10 | What could possibly go wrong? | 12 |
| 1.11 | The learning journey | 14 |
| 1.12 | Glossary | 14 |
| 1.13 | Exercises | 16 |
| 2 | Variables, expressions, and statements | 19 |
| 2.1 | Values and types | 19 |
| 2.2 | Variables | 20 |
| 2.3 | Variable names and keywords | 21 |
| 2.4 | Statements | 21 |
| 2.5 | Operators and operands | 22 |
| 2.6 | Expressions | 23 |
| 2.7 | Order of operations | 23 |
| 2.8 | Modulus operator | 24 |
| 2.9 | String operations | 24 |
| 2.10 | Asking the user for input | 24 |

| | | |
|----------|---|-----------|
| 2.11 | Comments | 25 |
| 2.12 | Choosing mnemonic variable names | 26 |
| 2.13 | Debugging | 28 |
| 2.14 | Glossary | 29 |
| 2.15 | Exercises | 30 |
| 3 | Conditional execution | 31 |
| 3.1 | Boolean expressions | 31 |
| 3.2 | Logical operators | 32 |
| 3.3 | Conditional execution | 32 |
| 3.4 | Alternative execution | 33 |
| 3.5 | Chained conditionals | 34 |
| 3.6 | Nested conditionals | 35 |
| 3.7 | Catching exceptions using try and except | 36 |
| 3.8 | Short-circuit evaluation of logical expressions | 37 |
| 3.9 | Debugging | 39 |
| 3.10 | Glossary | 39 |
| 3.11 | Exercises | 40 |
| 4 | Functions | 43 |
| 4.1 | Function calls | 43 |
| 4.2 | Built-in functions | 43 |
| 4.3 | Type conversion functions | 44 |
| 4.4 | Random numbers | 45 |
| 4.5 | Math functions | 46 |
| 4.6 | Adding new functions | 47 |
| 4.7 | Definitions and uses | 48 |
| 4.8 | Flow of execution | 49 |
| 4.9 | Parameters and arguments | 50 |
| 4.10 | Fruitful functions and void functions | 51 |
| 4.11 | Why functions? | 52 |
| 4.12 | Debugging | 52 |
| 4.13 | Glossary | 53 |
| 4.14 | Exercises | 54 |

| | | |
|----------|---|-----------|
| 5 | Iteration | 57 |
| 5.1 | Updating variables | 57 |
| 5.2 | The <code>while</code> statement | 57 |
| 5.3 | Infinite loops | 58 |
| 5.4 | “Infinite loops” and <code>break</code> | 58 |
| 5.5 | Finishing iterations with <code>continue</code> | 59 |
| 5.6 | Definite loops using <code>for</code> | 60 |
| 5.7 | Loop patterns | 61 |
| 5.7.1 | Counting and summing loops | 61 |
| 5.7.2 | Maximum and minimum loops | 62 |
| 5.8 | Debugging | 64 |
| 5.9 | Glossary | 64 |
| 5.10 | Exercises | 65 |
| 6 | Strings | 67 |
| 6.1 | A string is a sequence | 67 |
| 6.2 | Getting the length of a string using <code>len</code> | 68 |
| 6.3 | Traversal through a string with a loop | 68 |
| 6.4 | String slices | 69 |
| 6.5 | Strings are immutable | 70 |
| 6.6 | Looping and counting | 70 |
| 6.7 | The <code>in</code> operator | 71 |
| 6.8 | String comparison | 71 |
| 6.9 | <code>string</code> methods | 71 |
| 6.10 | Parsing strings | 74 |
| 6.11 | Format operator | 74 |
| 6.12 | Debugging | 75 |
| 6.13 | Glossary | 76 |
| 6.14 | Exercises | 78 |
| 7 | Files | 79 |
| 7.1 | Persistence | 79 |
| 7.2 | Opening files | 80 |
| 7.3 | Text files and lines | 81 |
| 7.4 | Reading files | 82 |

| | | |
|----------|--|------------|
| 7.5 | Searching through a file | 83 |
| 7.6 | Letting the user choose the file name | 85 |
| 7.7 | Using <code>try</code> , <code>except</code> , and <code>open</code> | 86 |
| 7.8 | Writing files | 87 |
| 7.9 | Debugging | 88 |
| 7.10 | Glossary | 88 |
| 7.11 | Exercises | 89 |
| 8 | Lists | 91 |
| 8.1 | A list is a sequence | 91 |
| 8.2 | Lists are mutable | 92 |
| 8.3 | Traversing a list | 92 |
| 8.4 | List operations | 93 |
| 8.5 | List slices | 94 |
| 8.6 | List methods | 94 |
| 8.7 | Deleting elements | 95 |
| 8.8 | Lists and functions | 96 |
| 8.9 | Lists and strings | 97 |
| 8.10 | Parsing lines | 98 |
| 8.11 | Objects and values | 99 |
| 8.12 | Aliasing | 100 |
| 8.13 | List arguments | 100 |
| 8.14 | Debugging | 102 |
| 8.15 | Glossary | 105 |
| 8.16 | Exercises | 106 |
| 9 | Dictionaries | 109 |
| 9.1 | Dictionary as a set of counters | 111 |
| 9.2 | Dictionaries and files | 112 |
| 9.3 | Looping and dictionaries | 113 |
| 9.4 | Advanced text parsing | 115 |
| 9.5 | Debugging | 116 |
| 9.6 | Glossary | 117 |
| 9.7 | Exercises | 118 |

| | |
|---|----------------|
| 10 Tuples | 119 |
| 10.1 Tuples are immutable | 119 |
| 10.2 Comparing tuples | 120 |
| 10.3 Tuple assignment | 122 |
| 10.4 Dictionaries and tuples | 123 |
| 10.5 Multiple assignment with dictionaries | 124 |
| 10.6 The most common words | 125 |
| 10.7 Using tuples as keys in dictionaries | 126 |
| 10.8 Sequences: strings, lists, and tuples - Oh My! | 126 |
| 10.9 Debugging | 127 |
| 10.10Glossary | 128 |
| 10.11Exercises | 129 |
| 11 Regular expressions | 131 |
| 11.1 Character matching in regular expressions | 132 |
| 11.2 Extracting data using regular expressions | 133 |
| 11.3 Combining searching and extracting | 136 |
| 11.4 Escape character | 139 |
| 11.5 Summary | 140 |
| 11.6 Bonus section for Unix / Linux users | 141 |
| 11.7 Debugging | 141 |
| 11.8 Glossary | 142 |
| 11.9 Exercises | 143 |
| 12 Networked programs | 145 |
| 12.1 HyperText Transport Protocol - HTTP | 145 |
| 12.2 The World's Simplest Web Browser | 146 |
| 12.3 Retrieving an image over HTTP | 147 |
| 12.4 Retrieving web pages with <code>urllib</code> | 150 |
| 12.5 Parsing HTML and scraping the web | 151 |
| 12.6 Parsing HTML using regular expressions | 151 |
| 12.7 Parsing HTML using BeautifulSoup | 152 |
| 12.8 Reading binary files using <code>urllib</code> | 154 |
| 12.9 Glossary | 155 |
| 12.10Exercises | 156 |

| | |
|--|------------|
| 13 Using Web Services | 157 |
| 13.1 eXtensible Markup Language - XML | 157 |
| 13.2 Parsing XML | 158 |
| 13.3 Looping through nodes | 159 |
| 13.4 JavaScript Object Notation - JSON | 160 |
| 13.5 Parsing JSON | 160 |
| 13.6 Application Programming Interfaces | 161 |
| 13.7 Google geocoding web service | 163 |
| 13.8 Security and API usage | 165 |
| 13.9 Glossary | 169 |
| 13.10Exercises | 170 |
| 14 Object-Oriented Programming | 171 |
| 14.1 Managing Larger Programs | 171 |
| 14.2 Getting Started | 172 |
| 14.3 Using Objects | 172 |
| 14.4 Starting with Programs | 173 |
| 14.5 Subdividing a Problem - Encapsulation | 175 |
| 14.6 Our First Python Object | 176 |
| 14.7 Classes as Types | 178 |
| 14.8 Object Lifecycle | 179 |
| 14.9 Many Instances | 180 |
| 14.10Inheritance | 181 |
| 14.11Summary | 182 |
| 14.12Glossary | 183 |
| 15 Using databases and SQL | 185 |
| 15.1 What is a database? | 185 |
| 15.2 Database concepts | 185 |
| 15.3 Database Browser for SQLite | 186 |
| 15.4 Creating a database table | 186 |
| 15.5 Structured Query Language summary | 189 |
| 15.6 Spidering Twitter using a database | 191 |
| 15.7 Basic data modeling | 196 |
| 15.8 Programming with multiple tables | 197 |

| | | |
|-----------|---|------------|
| 15.8.1 | Constraints in database tables | 200 |
| 15.8.2 | Retrieve and/or insert a record | 201 |
| 15.8.3 | Storing the friend relationship | 201 |
| 15.9 | Three kinds of keys | 203 |
| 15.10 | Using JOIN to retrieve data | 203 |
| 15.11 | Summary | 206 |
| 15.12 | Debugging | 206 |
| 15.13 | Glossary | 206 |
| 16 | Visualizing data | 209 |
| 16.1 | Building a Google map from geocoded data | 209 |
| 16.2 | Visualizing networks and interconnections | 211 |
| 16.3 | Visualizing mail data | 214 |
| A | Contributions | 221 |
| A.1 | Contributor List for Python for Everybody | 221 |
| A.2 | Contributor List for Python for Informatics | 221 |
| A.3 | Preface for “Think Python” | 221 |
| A.3.1 | The strange history of “Think Python” | 221 |
| A.3.2 | Acknowledgements for “Think Python” | 223 |
| A.4 | Contributor List for “Think Python” | 223 |
| B | Copyright Detail | 225 |