preface

Why write Python?

Python is an excellent, general-purpose programming language. You can write a program to send secret messages to your friends or to play chess. There are Python modules to help you wrangle complex scientific data, explore machine learning algorithms, and generate publication-ready graphics. Many college-level computer science programs have moved away from languages like C and Java to Python as their introductory language because Python is a relatively easy language to learn. We can use Python to study fundamental and powerful ideas from computer science. As I show you ideas like regular expressions and higher-order functions, I hope to encourage you to study further.

Why did I write this book?

Over the years, I've had many opportunities to help people learn programming, and I always find it rewarding. The structure of this book comes from my own experience in the classroom, where I think formal specifications and tests can be useful aids in learning how to break a program into smaller problems that need to be solved to create the whole program.

The biggest barrier to entry I've found when I'm learning a new language is that small concepts of the language are usually presented outside of any useful context. Most programming language tutorials will start with printing "HELLO, WORLD!" (and this is book is no exception). Usually that's pretty simple. After that, I usually struggle to write a complete program that will accept some arguments and do something *useful*.

In this book, I'll show you many, many examples of programs that do useful things, in the hopes that you can modify these programs to make more programs for your own use.

More than anything, I think you need to practice. It's like the old joke: "What's the way to Carnegie Hall? Practice, practice, practice." These coding challenges are short enough that you could probably finish each in a few hours or days. This is more material than I could work through in a semester-long university-level class, so I imagine the whole book will take you several months. I hope you will solve the problems, then think about them, and then return later to see if you can solve them differently, maybe using a more advanced technique or making them run faster.