1: INTRODUCTION

1.1: WHY SHOULD YOU LEARN TO WRITE PROGRAMS?

Writing programs (or programming) is a very creative and rewarding activity. You can write programs for many reasons, ranging from making your living to solving a difficult data analysis problem to having fun to helping someone else solve a problem. This book assumes that everyone needs to know how to program, and that once you know how to program you will figure out what you want to do with your newfound skills.

1.2: CREATIVITY AND MOTIVATION

Building useful, elegant, and clever programs for others to use is a very creative activity. Your computer or Personal Digital Assistant (PDA) usually contains many different programs from many different groups of programmers, each competing for your attention and interest. They try their best to meet your needs and give you a great user experience in the process. In some situations, when you choose a piece of software, the programmers are directly compensated because of your choice.

1.3: COMPUTER HARDWARE ARCHITECTURE

Before we start learning the language we speak to give instructions to computers to develop software, we need to learn a small amount about how computers are built.

1.4: UNDERSTANDING PROGRAMMING

In the rest of this book, we will try to turn you into a person who is skilled in the art of programming. In the end you will be a programmer - perhaps not a professional programmer, but at least you will have the skills to look at a data/information analysis problem and develop a program to solve the problem.

1.5: WORDS AND SENTENCES

Unlike human languages, the Python vocabulary is actually pretty small. We call this "vocabulary" the "reserved words". These are words that have very special meaning to Python. When Python sees these words in a Python program, they have one and only one meaning to Python. Later as you write programs you will make up your own words that have meaning to you called variables.

1.6: CONVERSING WITH PYTHON

Now that we have a word and a simple sentence that we know in Python, we need to know how to start a conversation with Python to test our new language skills.

1.7: TERMINOLOGY - INTERPRETER AND COMPILER

Python is a high-level language intended to be relatively straightforward for humans to read and write and for computers to read and process. Other high-level languages include Java, C++, PHP, Ruby, Basic, Perl, JavaScript, and many more. The actual hardware inside the Central Processing Unit (CPU) does not understand any of these high-level languages.

1.8: WRITING A PROGRAM

Typing commands into the Python interpreter is a great way to experiment with Python's features, but it is not recommended for solving more complex problems.

1.9: WHAT IS A PROGRAM?

The definition of a program at its most basic is a sequence of Python statements that have been crafted to do something. Even our simple hello.py script is a program. It is a one-line program and is not particularly useful, but in the strictest definition, it is a Python program. It might be easiest to understand what a program is by thinking about a problem that a program might be built to solve, and then looking at a program that would solve that problem.

1.10: THE BUILDING BLOCKS OF PROGRAMS

In the next few chapters, we will learn more about the vocabulary, sentence structure, paragraph structure, and story structure of Python. We will learn about the powerful capabilities of Python and how to compose those capabilities together to create useful programs. There are some low-level conceptual patterns that we use to construct programs. These constructs are not just for Python programs, they are part of every programming language from machine language up to the high-level languages.

1.11: WHAT COULD POSSIBLY GO WRONG?

As we saw in our earliest conversations with Python, we must communicate very precisely when we write Python code. The smallest deviation or mistake will cause Python to give up looking at your program. Beginning programmers often take the fact that Python leaves no room for errors as evidence that Python is mean, hateful, and cruel. While Python seems to like everyone else, Python knows them personally and holds a grudge against them.

1.12: THE LEARNING JOURNEY

As you progress through the rest of the book, don't be afraid if the concepts don't seem to fit together well the first time. When you were learning to speak, it was not a problem for your first few years that you just made cute gurgling noises. And it was OK if it took six months for you to move from simple vocabulary to simple sentences and took 5-6 more years to move from sentences to paragraphs, and a few more years to be able to write an interesting complete short story on your own.

1.E: INTRODUCTION (EXERCISES)

1.G: INTRODUCTION (GLOSSARY)