

# Preface

I have been teaching Python to first-year engineering students in Singapore University of Technology and Design. As a powerful and high level programming language, Python is ideal for the introduction to programming, which is useful for all engineering disciplines. In my teaching, I constantly find the need for ways to make my explanations as simple as possible. A concept, no matter how simple it seems to us computer scientists, may raise profound questions among students. For example, why is  $x = x + 1$  a valid statement but not equivalent to the contradicting equation  $0 = 1$ ? How can three lines of static code lead to a thousand statements being executed dynamically?

A student who asks such questions can be a very good student. Once the basic concepts are explained, she can manage programming effectively. On the other hand, few books on Python touch on the simple yet fundamental discussions that have been accumulated into my lecture notes. This has become my motivation for writing this book, a very gentle introduction to Python and programming.

From teaching experiences in University of Oxford, University of Cambridge, and Singapore University of Technology and Design, I find three components the most important for teaching programming. The first is syntax, the basic knowledge of how to write a correct program. As with most books on the Python language, it is an important component of this book. The second is the underlying mechanism of dynamic execution, and the underlying representation of data types. This is an aspect of programming that can easily be overlooked, which leads to common mistakes in programming. I make abstractions to the most important parts of the Python kernel, such as the binding table, object structures, and statement execution, showing them in figures, so that students can be confident of what happens underneath when a program is executed.

The third component is problem solving, which is the target of program design. The goal of learning programming is to solve problems, and it is important to associate programming concepts to problem solving when they are introduced. There are often various typical ways to approach common problems, and I try to introduce them along with the introduction to the programming language itself.

In addition, I include the most basic concepts in computer science into the introduction to Python, including information theory, computer architecture, data structure and algorithms, numerical analysis, and program design thinking such as functional programming and object-oriented programming. All these concepts are necessary backgrounds on which Python operates, and through which students can acquire a basic idea of computer science. I try to merge them into relevant sections, and make my introduction most simple and gentle.

In terms of Python syntax and libraries, I prioritize the former over the latter, introducing the full Python syntax and the most important libraries. The goal is to introduce Python to students with no programming or computer science background, equipping them with knowledge of the programming language and its underlying mechanisms, so that they can learn the usage of additional libraries with little difficulty, by consulting the Python documentation.

It took me over a year to transform my lecture notes into the book, during which many people gave me valuable help. Special thanks to Qiuqing Xu, Likun Qiu, Chen Lü, Yanan Lu, Chingyun Chang, Yijia Liu, Zhongye Jia, Bo Chen, Jie Yang, Meishan Zhang, Xiao Ding, Hao Zhou, and Ji Ma for helping me to draw the figures, and enter handwritten text in some chapters, to Haoliang Qi for inviting me to give a guest lecture series and encouragement to write up my lecture notes, and to all the students who gave me valuable feedback.

Yue Zhang

An Introduction to Python and Computer Programming

Zhang, Y.

2015, X, 295 p. 58 illus., 5 illus. in color., Hardcover

ISBN: 978-981-287-608-9