temario

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?

He estado enseñando Python a estudiantes de primer año de ingeniería en la Universidad de Tecnología y Diseño de Singapur. Como lenguaje de programación potente y de alto nivel, Python es ideal para la introducción a la programación, lo cual es útil para todas las disciplinas de ingeniería. En mi enseñanza, encuentro constantemente la necesidad de encontrar formas de hacer que mis explicaciones sean lo más simples posible. Un concepto, por simple que nos parezca a los informáticos, puede plantear profundas dudas entre los estudiantes. Por ejemplo, ¿por qué x = x + 1 es un enunciado válido pero no equivalente a la ecuación contradictoria 0 = 1? ¿Cómo es posible que tres líneas de código estático conduzcan a la ejecución dinámica de mil declaraciones?

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.

Un estudiante que hace estas preguntas puede ser un muy buen estudiante. Una vez explicados los conceptos básicos, podrá gestionar la programación de forma eficaz. Por otro lado, pocos libros sobre Python abordan las discusiones simples pero fundamentales que se han acumulado en mis notas de clase. Esta se ha convertido en mi motivación para escribir este libro, una introducción muy suave a Python y la programación.

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.

A partir de experiencias docentes en la Universidad de Oxford, la Universidad de Cambridge y la Universidad de Tecnología y Diseño de Singapur, encuentro tres componentes como los más importantes para la enseñanza de la programación. El primero es la sintaxis, el conocimiento básico de cómo escribir un programa correcto. Como ocurre con la mayoría de los libros sobre el lenguaje Python, es un componente importante de este libro. El segundo es el mecanismo subyacente de ejecución dinámica y la representación subyacente de los tipos de datos. Este es un aspecto de la programación que puede pasarse por alto fácilmente, lo que conduce a errores comunes en la programación. Hago abstracciones de las partes más importantes del kernel de Python, como la tabla de vinculación, las estructuras de objetos y la ejecución de declaraciones, mostrándolas en figuras, para que los estudiantes puedan estar seguros de lo que sucede debajo cuando se ejecuta un programa.

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.

El tercer componente es la resolución de problemas, que es el objetivo del diseño del programa. El objetivo de aprender programación es resolver problemas y es importante asociar los conceptos de programación con la resolución de problemas cuando se presentan. A menudo hay varias formas típicas de abordar problemas comunes y trato de presentarlas junto con la introducción al lenguaje de programación en sí.

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.

Además, incluyo los conceptos más básicos de la informática en la introducción a Python, incluida la teoría de la información, la arquitectura de la computadora, la estructura de datos y los algoritmos, el análisis numérico y el pensamiento de diseño de programas, como la programación funcional y la programación orientada a objetos. Todos estos conceptos son antecedentes necesarios sobre los que opera Python y a través de los cuales los estudiantes pueden adquirir una idea básica de la informática. Intento fusionarlos en secciones relevantes y hacer que mi introducción sea lo más sencilla y amable posible.

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.

En términos de sintaxis y bibliotecas de Python, priorizo las primeras sobre las segundas, introduciendo la sintaxis completa de Python y las bibliotecas más importantes. El objetivo es presentar Python a estudiantes sin experiencia en programación o informática, dotándolos de conocimientos sobre el lenguaje de programación y sus mecanismos subyacentes, para que puedan aprender el uso de bibliotecas adicionales con poca dificultad, consultando la documentación de Python.

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 Qiujing 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

Me llevó más de un año transformar mis apuntes de clase en el libro, durante el cual muchas personas me brindaron una valiosa ayuda. Un agradecimiento especial a Qiujing Xu, Likun Qiu, Chen Lü, Yanan Lu, Chingyun Chang, Yijia Liu, Zhongye Jia, Bo Chen, Jie Yang, Meishan Zhang, Xiao Ding, Hao Zhou y Ji Ma por ayudarme a dibujar las figuras. e ingresar texto escrito a mano en algunos capítulos, a Haoliang Qi por invitarme a dar una serie de conferencias como invitado y alentarme a escribir mis notas de clase, y a todos los estudiantes que me brindaron comentarios valiosos. Yue Zhang

Contents

**1 An Introduction to Python and Computer Programming . 1**

1.1 Introduction 1

1.1.1 Python and Computer Programming . 2

1.2 Preliminaries . 2

1.2.1 The Computer 3

1.2.2 The File System . 3

1.2.3 Text User Interfaces to Operating Systems . 5

1.2.4 The Python Application Program . 8

1.2.5 Python and Environment Variables 9

**2 Using Python as a Calculator . 13**

2.1 Using Python as a Calculator 13

2.1.1 Floating Point Expressions 15

2.1.2 Identifiers, Variables and Assignment 18

2.2 The Underlying Mechanism . 21

2.2.1 Information 22

2.2.2 Python Memory Management 25

2.3 More Mathematical Functions Using the math

and cmath Modules . 29

2.3.1 Complex Numbers and the cmath Module . 31

2.3.2 Random Numbers and the random Module 34

**3 The First Python Program . 37**

3.1 Text Input and Output Using Strings . 37

3.1.1 Text IO . 45

3.2 The First Python Program 49

3.2.1 The Structure of Python Programs 51

vii

3.3 The Underlying Mechanism of Module Execution 53

3.3.1 Module Objects . 54

3.3.2 Library Modules . 55

3.3.3 The Mechanism of Module Importation . 56

3.3.4 Duplicated Imports . 58

3.3.5 Importing Specific Identifiers 60

**4 Branching and Looping . 67**

4.1 The Boolean Type . 68

4.2 Branching Using the if Statement . 72

4.2.1 Nested if Statements 78

4.3 Looping Using the While Statement . 81

4.3.1 Branching Nested in a Loop . 86

4.3.2 Break and Continue 88

4.4 Debugging . 89

**5 Problem Solving Using Branches and Loops 97**

5.1 Basic Problems 97

5.1.1 Summation . 97

5.1.2 Iteratively Calculating Number Sequences . 102

5.2 Numerical Analysis Problems 105

5.2.1 Numerical Differentiation . 105

5.2.2 Numerical Integration . 106

5.2.3 Monte-Carlo Methods . 109

5.2.4 Differential Equations and Iterative Root Finding 113

5.3 Tuples and the for loop 116

5.3.1 Tuples 116

5.3.2 The for Loop . 120

5.3.3 Problem Solving by Traversal of a Tuple 122

**6 Functions 127**

6.1 Function Definition Using lambda expressions 127

6.2 Function Definition Using the def Statement . 132

6.2.1 The Dynamic Execution Process of Function Calls . 135

6.2.2 Input Arguments . 136

6.2.3 Return Statements 137

6.2.4 Modularity . 140

6.3 Identifier Scopes . 144

6.4 The Underlying Mechanism of Functions 148

**7 Lists and Mutability 157**

7.1 Lists—A Mutable Sequential Type 157

7.1.1 List Mutation . 160

viii Contents

7.2 Working with Lists . 166

7.2.1 Copying Lists . 167

7.2.2 Lists as Items in Tuples and Lists . 169

7.2.3 Lists and Loops . 173

7.2.4 Lists and Function Arguments . 177

7.2.5 Lists and Function Return Values . 178

7.2.6 Initializing a List 180

7.2.7 Lists and Sequential Data Structures . 181

**8 Sequences, Mappings and Sets . 187**

8.1 Methods of Sequential Types 187

8.2 Dicts—A Mutable Mapping Type . 195

8.2.1 Dict Modification 199

8.2.2 Dicts and Loops . 201

8.2.3 Dicts and Functions 203

8.3 Sets and Bitwise Operations . 205

8.3.1 Set Modification . 207

8.3.2 Bitsets and Bitwise Operators 209

**9 Problem Solving Using Lists and Functions . 217**

9.1 Lists of Lists and Nested Loops 217

9.1.1 Treating Sublists as Atomic Units . 217

9.1.2 Matrices as Lists of Lists . 221

9.2 Functions and Problem Solving 224

9.2.1 Recursive Function Calls . 225

9.2.2 Functional Programming . 229

9.3 Files, Serialization and urllib 236

9.3.1 Files . 236

9.3.2 Serialization Using the pickle Module 240

9.3.3 Reading Web Pages Using the urllib Module . 241

**10 Classes 245**

10.1 Classes and Instances . 246

10.1.1 Classes and Attributes . 246

10.1.2 Methods and Constructors 248

10.1.3 Class Attributes and the Execution

of a Class Statement 252

10.1.4 Special Methods . 253

10.1.5 Class Examples . 257

10.1.6 The Underlying Mechanism of Classes

and Instances . 260

10.2 Inheritance and Object Oriented Programming 263

10.2.1 Sub Classes 264

10.2.2 Overriding Methods 266

Contents ix

10.2.3 The Underlying Mechanism of Class Extention . 267

10.2.4 Object Oriented Programming . 269

10.3 Exception Handling . 269

10.3.1 Exception Handling . 271

10.3.2 Exception Objects 274

**11 Summary 279**

11.1 The Structure of a Python Program 279

11.1.1 Expressions 279

11.1.2 Statements . 283

11.2 The Data Model of Python 285

11.2.1 Identity, Type and Value . 285

11.2.2 Attributes and Methods 286

11.2.3 Documenting Objects . 287

11.3 Modules and Libraries . 289

11.3.1 Packages 290

11.3.2 Library Modules .