



H3C 大数据人才培养规划教材

Python

程序设计基础

Python Programming

BIG
DATA
TECHNOLOGY

蔡永铭 / 主编 熊伟 / 副主编 林子雨 / 主审

- ◆ 突出对学生的计算思维训练
- ◆ 通过几个大案例详细讲解从设计到开发全过程
- ◆ 免费赠送全部实例的源代码及课件

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS



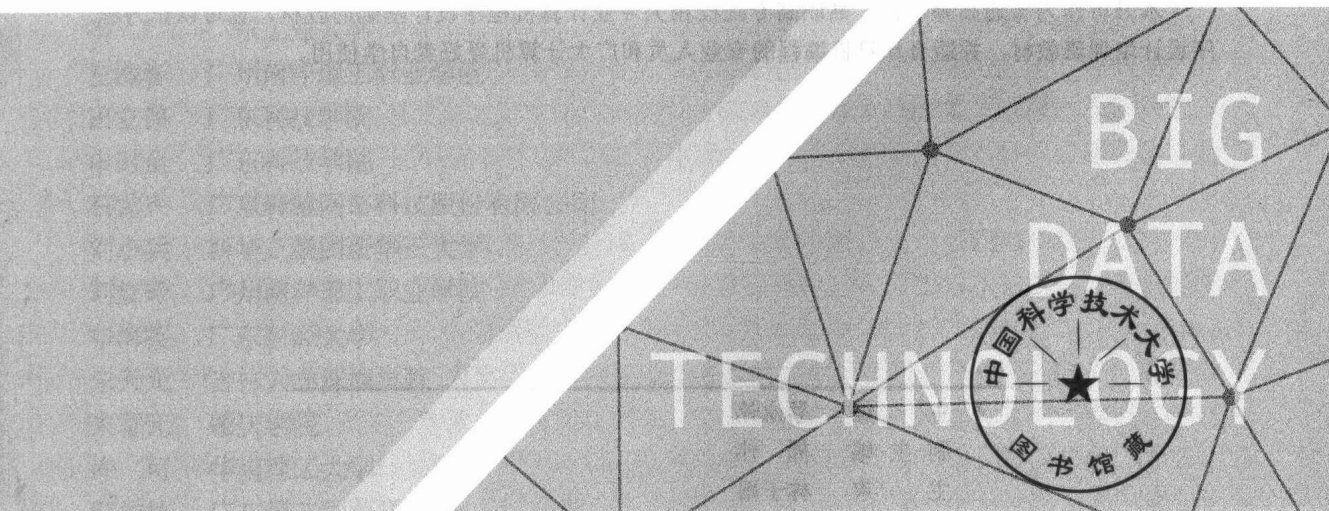
H3C 大数据人才培养规划教材



Python

程序设计基础

Python Programming



蔡永铭 / 主编 熊伟 / 副主编 林子雨 / 主审

RFID

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Python程序设计基础 / 蔡永铭主编. -- 北京 : 人民邮电出版社, 2019. 1
大数据人才培养规划教材
ISBN 978-7-115-49015-5

I. ①P… II. ①蔡… III. ①软件工具—程序设计—教材 IV. ①TP311.56

中国版本图书馆CIP数据核字(2018)第174779号

内 容 提 要

本书较为全面地介绍 Python 程序设计基础。全书共 13 章, 主要包括 Python 简介、基础语法知识、条件、循环、字符串、文件、列表、元组、函数、字典、异常和异常处理、图形用户界面、面向对象程序设计、数据库支持、程序开发进阶等。每章后面都提供习题和实战作业, 通过练习和操作实践, 帮助读者巩固所学的内容。

本书可作为普通高等学校、高职高专院校相关专业计算机程序设计基础的教材, 也可以作为程序设计培训班教材, 并适合计算机编程的专业人员和广大计算机爱好者自学使用。

◆ 主 编 蔡永铭
副 主 编 熊 伟
主 审 林子雨
责任编辑 马小霞
责任印制 马振武

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16
印张: 15.5 2019 年 1 月第 1 版
字数: 398 千字 2019 年 1 月河北第 1 次印刷

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

丛书编委会

主任：

林子雨 厦门大学

委员：

万君社 广州沐祥软件

于 鹏 新华三集团新华三大学

王正勤 广州南洋理工职业学院

王会林 韩山师范学院

王国华 华南理工大学

王敏琴 肇庆学院

冯 广 广东工业大学

左海春 广州南洋理工职业学院

田立伟 广东科技学院

申时全 广东科技学院

石龙兴 广东轩辕网络科技股份有限公司

刘小兵 新华三集团新华三大学

刘红玲 广州南洋理工职业学院

华海英 广东财经大学

朱天元 吉林大学珠海学院

朱香元 肇庆学院

许 可 华南理工大学

邬依林 广东第二师范学院

余姜德 中山职业技术学院

张小波 广东轩辕网络科技股份有限公司

张 纯 汕头广播电视大学

张建明 长沙理工大学

张艳红 广东技术师范学院天河学院

李 舒 中国医科大学

杨胜利 广东科技学院

杨 峰 广东财经大学

苏 绚 汕头广播电视大学

邱 新 汕头广播电视大学

陈永波 新华三集团新华三大学

陈 统 广东轩辕网络科技股份有限公司
陈 强 广东科技学院
周永塔 广东南华工商职业学院
周永福 河源职业技术学院
周 奇 广东开放大学
罗定福 广东松山职业技术学院
郑海清 广东南华工商职业学院
柳义筠 广州科技贸易职业学院
贺敏伟 广东财经大学
翁 健 暨南大学
黄清宝 广西大学
龚旭辉 广东工业大学
曾振东 广东青年职业学院
谢 锐 广东工业大学
简碧园 广州科技职业技术学院
廖大强 广东南华工商职业学院
熊 伟 广东药科大学
蔡永铭 广东药科大学
蔡 毅 华南理工大学

序

Preface

从2010年开始,人类正式步入了大数据时代。现在,大数据已经在各行各业得到了广泛的应用,产生了巨大的经济和社会价值。很多企业纷纷部署大数据分析平台,利用数据来驱动企业生产和运营,市场上对数据科学与大数据人才的需求日益旺盛。

为了满足市场对大数据人才的迫切需求,高校肩负起了大数据专业人才培养的重任。清华大学、北京大学、厦门大学、复旦大学、中国人民大学等一批高校在国内率先开设大数据课程;2016年,北京大学、中南大学、对外经济贸易大学三所高校成为国内首批获得教育部批准设立“数据科学与大数据技术”专业的本科院校。此后,教育部又于2017年和2018年分别批准32所和248所本科院校设立“数据科学与大数据技术”专业。与此同时,根据教育部公布的“大数据技术与应用”专业备案和审批结果,截至2018年5月,已经有累计208所职业院校获批“大数据技术与应用”专业。随着大数据专业在国内众多高校的开设,大数据专业人才的培养进入了全新的阶段。

大数据专业的开设,只是吹响了国内高校开展大规模大数据人才培养的冲锋号角。接下来,全国高校的大数据专业建设者,就要面对专业建设中必须要解决的专业建设方案、课程体系、教材、师资等一系列问题。这其中,课程教材和师资队伍的建设,是决定大数据这个新兴专业能否达成预期人才培养目标的关键。

作为全国较早探索大数据教学的教师之一,本人编著了国内高校第一本系统性介绍大数据知识的专业教材《大数据技术原理与应用》,成为国内众多高校的开课教材,同时,带领厦门大学数据库实验室团队建设了国内高校首个大数据课程公共服务平台,为全国高校大数据教学提供一站式服务,平台每年访问量超过200万次,成为全国高校大数据教学知名品牌。但是,我们团队的工作放在全国高校大数据专业发展的全局中考量,是微不足道的。为了满足复合型大数据人才的培养要求,在大数据专业课程体系的设计中,包含了数据采集、数据清洗、数据存储与管理、数据挖掘、数据分析、数据可视化等一系列课程。这些课程需要一系列的教材。正是在这样一个大背景之下,本套大数据系列教材应运而生。同时,我也很荣幸,能够成为本套教材编委会的主任。

本套大数据系列教材,在教材选题阶段,就经过了系统、严谨的论证,来自高校、大数据企业、出版社的多位专家学者,多次举行现场论证会和网络沟通会,在教材的目标定位、知识体系和内容编排等方面交换意见,形成共识。通过选题论证,明确了系列教材的总体知识框架,梳理了每本教材的写作大纲和知识要点,为最终形成高质量的系列教材奠定了坚实的基础。在教材写作阶段,本套教材的作者认真遵守系列教材的总体写作规范,努力做到教材知识结构清晰、内容编排合理、写作详略得当。作为本套教材编委会主任,我对每本教材的初稿都进行了认真审核,严把文字质量关,并提出有针对性的修改意见。编委会、教材作者、出版社等多方面的认真、严谨工作,是促成这套高质量大数据教材诞生的重要保证。

作为经过整体规划的系列教材,本套大数据教材非常注重多本教材之间的知识布局,力求做到教材与教材之间的“知识分割不重复”和“知识传承不断层”,从而可以较好地满足高校大数据专业

建设过程中对于课程体系中配套教材的需求，让教师教课“有纲可循”，让学生学习“脉络清晰”。

教学工作需要大量的实践，教材水平需要在教学实践的反复检验中不断提升。本套教材的出版，不是教材创作的结束，而是一个新阶段的开始。当本套教材应用到各个高校的教学实践以后，难免会遇到各种各样的问题，但我也相信，有教材使用者积极反馈的意见，有教材编写者的努力认真修改，本套教材的质量和水平会不断得到提升；在全国高校大数据教学工作者的共同努力下，大数据专业的建设与发展会不断迈上新的台阶。

林子雨
于厦门大学数据库实验室
2018年8月

前言

Foreword

随着计算机专业与各个学科的高度交叉发展，高等院校的计算机基础课程已经不能仅仅局限于计算机基础知识与办公软件，而应把计算机作为一种工具，融入各个专业。为此，我们现在上计算机课，更应该在课程中体现出计算机科学的思维和方法，并且在教学的过程中突出对学生计算思维的训练。

Python 语言是一门发展了近 30 年的编程语言，是目前美国大学最受欢迎的程序设计语言之一，目前也越来越受到我国各大院校的重视。Python 语言实际内容是分支、循环、函数等基本的程序逻辑关系及功能强大的函数库应用。该语言只关心计算问题的求解，其轻量级的语法和高层次的语言表示展现了应用计算机解决问题的计算思维理念。它是目前最接近自然语言的通用编程语言。从解决计算问题角度来看，传统的 C、Java 和 VB 语言过分强调语法并不适合非计算机专业学生解决一般的计算问题，而 Python 这种轻语法的程序设计语言，让学生能更好地从语法的学习状态中解放出来，从而有更多的时间来解决所面临的各类计算问题。

全书共 13 章，较为全面地介绍了 Python 这门计算机语言。第 1 章主要对 Python 进行了简单介绍，第 2、3 章主要介绍了 Python 的编程基础知识、顺序、分支、循环结构。第 4、5、6、8、9 章主要介绍了 Python 基本的数据类型及数据结构，包括字符串、文件、列表、元组、字典、异常等。第 7 章是函数，对于读者来说，这一章是比较抽象的，但也是重点。对于学习编程的人来说，没有掌握函数就像没有掌握编程的核心一样。第 10、11、12 章是图形用户界面、面向对象程序设计、数据库支持等内容，主要是针对学有余力的读者的。由于篇幅有限，不可能把这些内容都介绍得很详细，有兴趣的读者可以通过查阅相关资料，进一步深入学习。第 13 章是程序开发进阶，作者设计了几个大的案例，详细讲解了从设计到开发的全过程，让读者全面了解计算机科学的思维与方法，并学习解决面临的实际问题。另外，每章后面都提供了一些习题和实战作业，通过练习和操作实践，帮助读者巩固所学的内容。本书通俗易懂，每一章都有一个引例进行引入，适合普通高等学校、高职高专计算机专业及非计算机专业的学生阅读；对于编程爱好者来说，也是一个不错的选择。

为方便读者使用，本书全部实例的源代码及课件均免费赠送给读者，可登录人民邮电出版社人邮教育社区（www.ryjiaoyu.com）下载。

本书由蔡永铭担任主编，熊伟担任副主编，林子雨担任主审。其他参与编写工作的还有黄国权、王胜、周苏娟、余珊珊、曾小燕、刘珍、郑建华等老师。

由于编者水平和经验有限，书中难免有欠妥和疏漏之处，恳请读者批评指正。

编者 E-mail 为 148157@qq.com。

编者

2018 年 8 月

目录

Contents

第 1 章 Python 简介	1		
1.1 认识 Python	1		
1.1.1 什么是 Python 语言	1		
1.1.2 Python 语言的优点与缺点	2		
1.1.3 为什么选择 Python 语言	3		
1.2 Python 的安装	4		
1.2.1 Windows 环境下安装	4		
1.2.2 Linux 和 UNIX 环境下安装	6		
本章小结	7		
练习题	7		
实战作业	7		
第 2 章 基础语法知识	8		
2.1 数字和表达式	8		
2.2 变量	10		
2.3 语句	11		
2.4 获取用户输入	11		
2.5 函数	12		
2.6 模块	13		
2.7 程序的运行	14		
2.7.1 通过命令提示符运行 Python 脚本	14		
2.7.2 让脚本像普通程序一样运行	14		
2.7.3 注释	15		
2.8 字符串	16		
2.8.1 单引号字符串和转义引号	16		
2.8.2 拼接字符串	17		
2.8.3 字符串 str 和 repr	18		
2.8.4 input	19		
2.8.5 长字符串、原始字符串和 Unicode	19		
本章小结	21		
练习题	22		
第 3 章 条件、循环和其他语句	23		
3.1 print 语句的应用	23		
3.2 赋值语句	24		
3.2.1 序列解包	25		
3.2.2 链式赋值	25		
3.2.3 增量赋值	25		
3.3 代码块与缩进	26		
3.4 条件语句	26		
3.4.1 布尔变量	26		
3.4.2 if 语句	26		
3.4.3 else 子句	27		
3.4.4 elif 子句	27		
3.4.5 嵌套条件语句	29		
3.5 循环语句	30		
3.5.1 while 循环	31		
3.5.2 for 循环	32		
3.5.3 Python 循环嵌套	34		
3.5.4 跳出循环 (break 与 continue)	34		
3.5.5 循环中的 else 子句	35		
3.5.6 综合应用	36		
本章小结	38		
练习题	38		
实战作业	41		
第 4 章 字符串	43		
4.1 字符串的基本操作	43		

7.3.2	关键字参数	108	8.5	字典的排序	135
7.3.3	默认参数	108	8.5.1	sorted 函数按 key 值 对字典排序	135
7.3.4	不定长参数	109	8.5.2	sorted 函数按 value 值 对字典排序	135
7.4	匿名函数	110	8.6	字典的实例应用	136
7.5	变量作用域	111	本章小结		140
7.6	函数的属性	113	练习题		140
7.7	Python 中的 main 函数	114	实战作业		140
7.8	Python 的 zip 函数	115			
7.9	常见内置函数	116			
7.9.1	与数学相关的函数	116			
7.9.2	类型转换函数	117			
7.9.3	相关操作函数	118			
7.10	函数的应用举例	119			
本章小结		124			
练习题		124			
实战作业		125			
第 8 章 字典		126	第 9 章 异常和异常处理		142
8.1	什么是字典	126	9.1	错误与异常	143
8.2	字典的创建和使用	127	9.1.1	语法错误	143
8.2.1	直接创建字典	127	9.1.2	异常	143
8.2.2	通过 dict 函数创建字典	128	9.2	异常处理	144
8.2.3	字典的修改和删除	128	9.2.1	异常处理语法	144
8.2.4	字典的遍历	129	9.2.2	常见异常处理示例	145
8.3	字典的基本操作	130	9.3	抛出异常和自定义异常	146
8.3.1	get 函数: 访问字典成员	130	9.3.1	如何抛出异常	146
8.3.2	copy 函数: 返回一个 具有相同键值的新字典	131	9.3.2	用户自定义异常	147
8.3.3	pop 函数: 删除字典中 对应的键和值	132	本章小结		148
8.3.4	Fromkeys 函数: 用给定的键 建立新的字典	132	练习题		148
8.3.5	setdefault 函数: 获取与 给定键相关联的值	132			
8.3.6	update 函数: 用一个字典 更新另外一个字典	133			
8.3.7	关键字 in 的应用	133			
8.4	字典的格式化字符串	134			
			第 10 章 图形用户界面		151
			10.1	丰富的平台	151
			10.2	下载和安装 wxPython	152
			10.3	创建示例 GUI 应用程序	152
			10.3.1	Hello World!	152
			10.3.2	窗口	154
			10.3.3	控件: 面板、标签、菜单栏	154
			10.3.4	案例制作: 简易编辑器	157
			10.3.5	更智能的布局	159
			10.3.6	最基本的 Sizer: GridSizer	160
			10.3.7	FlexGridSizer	162
			10.3.8	GridBagSizer	164
			10.3.9	BoxSizer	165
			10.3.10	StaticBoxSizer	168
			10.3.11	案例制作: 用户收件	

信息录入界面	169
10.4 初步了解 tkinter	172
本章小结	172
练习题	173
实战作业	173

第 11 章 面向对象程序设计 174

11.1 面向对象程序设计思想	175
11.1.1 面向对象的基本概念	175
11.1.2 面向对象设计	177
11.2 类和对象	177
11.2.1 创建类	177
11.2.2 创建对象	178
11.3 属性和数据	178
11.3.1 类属性	178
11.3.2 实例属性	179
11.3.3 类属性与实例属性的联系	180
11.3.4 私有属性与公有属性	180
11.3.5 自定义属性	181
11.3.6 self 的作用	182
11.4 方法	182
11.4.1 方法的声明和调用	182
11.4.2 实例方法、类方法和静态方法	183
11.4.3 绑定方法和非绑定方法	185
11.4.4 私有方法与公有方法	187
11.4.5 构造方法与析构方法	187
11.4.6 特殊方法	189
11.5 继承	189
11.5.1 概念	189
11.5.2 单继承	189
11.5.3 继承与抽象	190
11.5.4 覆盖方法	191
11.5.5 多重继承	192
11.6 多态和封装	193
11.6.1 多态性	193
11.6.2 封装和私有化	194
11.7 定制类	196

11.8 迭代器	198
11.9 生成器	199
11.10 综合示例	200
本章小结	202
练习题	202
实战作业	203

第 12 章 数据库支持 204

12.1 数据库概述	205
12.2 Python 数据库编程接口 (DB-API)	205
12.2.1 DB-API 简介与统一操作模式	205
12.2.2 模块接口 (Module Interface)	206
12.2.3 Connection 对象	207
12.2.4 Cursor 游标对象	208
12.3 MySQL 关系数据库操作	208
12.3.1 PyMySQL 下载与安装	209
12.3.2 PyMySQL 基本操作	209
12.4 SQLite 数据库与操作	212
12.4.1 SQLite 介绍与安装	212
12.4.2 SQLite-Python 接口及操作	214
本章小结	217
练习题	218
实战作业	218

第 13 章 程序开发进阶 219

13.1 简介	220
13.2 分治算法	220
13.3 鸮尾花的分类	221
13.3.1 KNN (K-NearestNeighbor) 算法介绍	221
13.3.2 K 个最近邻居	222
13.3.3 构造分类样本	222
13.4 设计分类算法	223
13.5 详细设计	224

13.5.1	文件格式	224	13.5.8	不足之处	232
13.5.2	存储格式	225	13.6	其他有趣的算法问题	232
13.5.3	loadDataset 函数	225	13.6.1	乳腺癌分类问题	232
13.5.4	getNeighbors 函数	226	13.6.2	葡萄酒品质分类	234
13.5.5	getResponse 函数	228	本章小结		235
13.5.6	getAccuracy 函数	229	练习题		235
13.5.7	总体运行	229	实战作业		235

第1章

Python简介



引例

我开始设计一种语言，使得程序员的效率更高。

——吉多·范罗苏姆（Python 语言设计者）

那么，什么是 Python？为什么要使用它？谁该使用它？让我们带着这些问题一起走进 Python。

1.1 认识 Python

1.1.1 什么是 Python 语言

Python（发音[ˈpaɪθən]），是一种面向对象的解释型计算机程序设计语言，由荷兰人吉多·范罗苏姆（Guido van Rossum）于1989年发明。Python的第一个公开发行人版发行于1991年。

Python是纯粹的自由软件，源代码和解释器CPython遵循GPL（GNU General Public License）协议。Python语法简洁清晰，特色之一是强制用空白符（white space）作为语句缩进。

Python具有丰富和强大的库。它常被称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地连接在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，如果性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。需要注意的是，在使用扩展类库时，可能需要考虑平台问题，某些扩展类库可能不提供跨平台的实现。

IEEE发布了2017年编程语言排行榜，Python高居首位。2000年Python发布了2.0版本，2008年Python发布了3.0版本。

1.1.2 Python 语言的优点与缺点

Python 作为一门高级编程语言，虽然诞生的时间并不很长，但是却得到了程序员的喜爱。Python 程序简单易懂，对于初学者而言，Python 很容易入门，而且随着学习的深入，学习者可以使用 Python 语言编写非常复杂的程序。但是，编程语言不可能是完美的，总有自己的优势与劣势，Python 也一样，也有自己的优缺点，下面就来梳理一下 Python 语言的优缺点。

1. Python 语言的优点

(1) 可使用多种执行方式

① 可以直接在命令行执行相关命令。

【例 1-1】 直接在命令行执行 print 打印命令。

```
>>> print('Hello,Python!')
Hello,Python!
>>> sum=99999*99999
>>> print(sum)
9999800001
```

② 可以用函数的方式执行相关命令。

【例 1-2】 自定义加法函数，并应用。

```
>>> def add(num1,num2):
    return num1+num2
>>> add(3,5)
8
```

③ 可以用面向对象的方式执行相关命令。

【例 1-3】 使用 turtle 对象画出公切线相同、大小不同的圆，如图 1-1 所示。

```
>>> import turtle
>>> turtle.pensize(2)
>>> turtle.circle(10)
>>> turtle.circle(40)
>>> turtle.circle(80)
>>> turtle.circle(120)
```

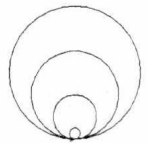


图 1-1 带有公切线的圆

(2) 语法简洁，且强制缩格

【例 1-4】 编程求 2000 年~2500 年的闰年。

```
1. i=2000
2. j=0
3. while i<=2500:
4.     if i %4==0 and i%100!=0 or i %400==0:
5.         print(i)
6.     j+=1
7.     if j%10==0:
8.         print
9.     i+=1
```

从程序上可以看出，该程序具有可读性及强制缩格的功能。

(3) 跨平台

支持多种开发平台，如 Windows、Linux、Mac OS X、Solaris……

(4) 开源开放

截至 2018 年，全球有 9 万多个登记的开源库，覆盖各类计算问题，且开源库的数量以 1 万/年左右逐年增加。

(5) 面向对象

Python 既支持面向过程，又支持面向对象，这使得其编程更加灵活。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如 C++ 和 Java 相比，Python 以一种非常强大又简单的方式实现面向对象编程。

(6) 丰富的第三方库

Python 有丰富而且强大的库，而且由于 Python 的开源特性，第三方库非常多，如 Web 开发、爬虫、科学计算等。

2. Python 语言的缺点

Python 虽然有很多优点，但也不是完美的，它也有自身的缺点。

(1) 速度慢。由于 Python 是解释型语言，所以它的速度会比 C、C++ 慢一些，但是不影响使用。由于现在的硬件配置都非常高，基本上没有影响，除非是一些实时性比较强的程序可能会受到一些影响，但是也有解决办法，比如可以嵌入 C 程序。

(2) 强制缩进。如果读者有其他语言的编程经验，如 C 语言或者 Java 语言，那么 Python 的强制缩进一开始会让你很不习惯。但是如果读者习惯了 Python 的缩进语法，就会觉得它非常优雅。

(3) 单行语句。由于 Python 在行尾可以不写分号，所以一行只能有一条语句。

1.1.3 为什么选择 Python 语言

1. 院校程序类课程的需要

计算机程序设计基础课是各院校类非计算机专业的必修课程。近年来，各大院校也都尝试使用各种程序设计语言来进行授课，其中不乏一些经典的程序设计语言，如 C、C++、VB、Java 等，但是对于教学效果来说，却乏善可陈。Python 语言是一种解释型、面向对象的计算机程序设计语言，广泛用于计算机程序设计教学语言、系统管理编程脚本语言、科学计算等，特别适用于快速的应用程序开发。目前，各大院校已经越来越重视 Python 教学，Python 已经成为最受欢迎的程序设计语言之一。

与计算机专业教学不同，面向非计算机专业的计算机基础类程序设计课程的定位应该是：通过某一编程语言的教学，传授利用计算思维解决一般计算问题的基本方法，并能够通过程序设计更好地利用计算机强大的计算性能。在这个技术时代更应关注问题的求解，超越对程序执行性能、代码高复用性或某一个特殊系统中视窗设计的关注，让学生真正掌握利用计算机解决计算问题的通用方法。

2. 高级语言发展的必然选择

从程序设计语言发展角度来看，高级编程语言的设计一直追求接近人类的自然语言。这样的高级语言也在不断进化，如 C、Java、VB 等，Python 语言则更进一步，提供十分接近人类理解的语法形式。应该说，Python 语言发展了高级语言的表达形式，简化了程序设计过程，提升了程序设计

效率。从计算思维培养角度来说,传统 C、Java 和 VB 语言过分强调语法,并不适合非计算机专业的学生。从传统应用技能教育向计算思维培养转变过程中,教学内容变革是重中之重。对于程序设计课程,选择适合技术时代发展的编程语言,是显著提高培养效果的前提和基础。从解决计算问题角度,传统 C、Java 和 VB 语言过分强调语法并不适合非计算机专业学生解决一般计算问题,而 Python 语言作为适应新技术时代的“轻语法”程序设计语言,已经得到大学计算机教育领域的重视。因此,Python 语言相比其他语言具有更高的教学价值。这一技术趋势也得到了国内外众多大学的直接响应。

随着大数据、云计算、网络空间安全等概念的兴起,当代信息社会要求大学生在具备操作计算机的基本技能外,还要具备一定的编程能力,用于解决工作和学习中遇到的各类非通用计算问题,理解并实践计算思维。这种信息时代深入发展的趋势为程序设计课程的内容改革提供了依据。程序设计课程教学内容的变化是一个正常的技术更迭过程。从 20 世纪 90 年代开始,程序设计课程的教学内容经过几次较大变化,曾经广泛用于教学的 Pascal 和 Fortran 语言,被 C、Java 和 VB 等语言取代。然而,随着大数据、云计算、物联网、信息安全等各种计算形态的高速发展,程序设计语言教学内容却还基本没有变化,针对非计算机专业学生主要开设 C 语言、Java 语言和 VB 语言。教学内容近十几年的稳定,并非因为上述教学内容达到了教学预期,而是受制于特定技术时代的历史局限性。

Python 语言的易学易用和丰富的开源库,将会给学生带来一个全新的程序设计认识,改变主观渴望学好编程语言但客观上学不会学不精的状况,有助于帮助学生学会一种终身受用的编程语言,进而帮助他们更好地利用计算机解决所面临的各类计算问题。Python 语言的高编写效率,会进一步加快程序实现和修改节奏,降低程序错误率,缩短计算服务和产品的上市时间,提高国民综合生产效率。在我国全面提高高等教育质量、广泛开展本科教学背景下,Python 语言教学改革将会开启一个全新的程序设计语言教学时代,进一步释放师生活力和创新热情,全面提升教学质量。以 Python 语言教学为手段,将更容易开展计算思维教育教学活动,使学生在思维和技能两方面终身受益。

1.2 Python 的安装

1.2.1 Windows 环境下安装

在开始编程前,需要先安装一些软件。下面简要介绍如何下载和安装 Python。如果想直接跳到安装过程的介绍而不看详细的向导,可以登录官网下载区,选择需要的版本,本书选择的是 3.4 的 Windows 64 位版本。

要在 Windows 下安装 Python,请参考以下步骤。

- (1) 下载好 Python 软件之后,单击 Python 的安装图标,就会出现图 1-2 所示的界面。
- (2) 安装时需要配置 Python 环境变量。首先找到 Python 的安装位置,一般系统都是默认在 C 盘安装的,如图 1-3 所示。
- (3) 用鼠标右键单击计算机图标,选择属性,进入“高级系统设置”界面。
- (4) 进入高级系统设置,如图 1-4 所示,单击“环境变量”按钮。
- (5) 在环境变量里的系统变量中,找到 path,选择编辑,在变量值后面添加“c:\python34”路径,注意要在这前面加上英文的分号(;),如图 1-5 所示。