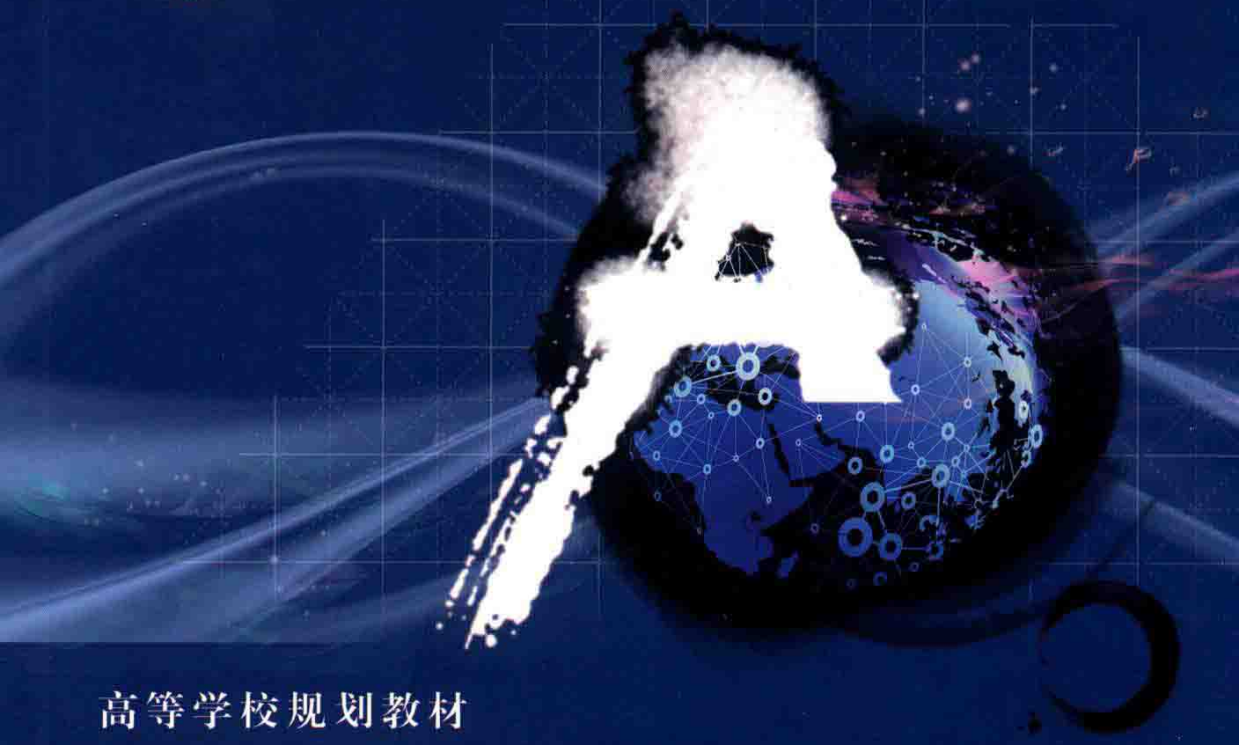
普通高等教育“十三五”规划教材



高等学校规划教材

Python大学教程

◎ 吕云翔　赵天宇　张　元　编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十三五”规划教材
高等学校规划教材

Python 大学教程

吕云翔　赵天宇　张　元　编　著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书介绍了使用 Python 语言进行程序设计的方法及其应用。

全书共 14 章，分为三部分。第一部分为基础篇（第 1～5 章），主要介绍 Python 的基础语法，包括 Python 语言的概述，Python 的基本数据类型、变量、运算符、表达式等概念，三种程序的基本控制结构，函数、列表、元组、集合与字典四种简单数据结构。第二部分为进阶篇（第 6～10 章），主要介绍 Python 的一些高级特性和功能，包括模块和包的使用，字符串的处理，Python 的面向对象编程，异常处理和文件处理。第三部分为应用篇（第 11～14 章），主要介绍 Python 在某些领域的应用方法，包括 GUI 程序开发、数据管理和数据库管理、Web 开发及多任务编程。

本书既可以作为高等院校计算机与软件相关专业的教材，也可以作为软件从业人员的学习指导用书。

图书在版编目（CIP）数据

Python 大学教程 / 吕云翔，赵天宇，张元编著. —北京：电子工业出版社，2017.9

ISBN 978-7-121-31944-0

Ⅰ. ①P… Ⅱ. ①吕… ②赵… ③张… Ⅲ. ①软件工具—程序设计—高等学校—教材 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2017）第 139815 号

策划编辑：戴晨辰
责任编辑：裴　杰
印　　刷：天津嘉恒印务有限公司
装　　订：天津嘉恒印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1 092　1/16　印张：16.5　字数：486 千字
版　　次：2017 年 9 月第 1 版
印　　次：2017 年 9 月第 1 次印刷
定　　价：42.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：dcc@phei.com.cn。

前　言

Python 是一种解释型、支持面向对象特性的、动态数据类型的高级程序设计语言。自从 20 世纪 90 年代 Python 公开发布以来，经过二十多年的发展，Python 以其语法简洁而高效、类库丰富而强大、适合快速开发等原因，成为当下最流行的脚本语言之一，也广泛应用到了统计分析、计算可视化、图像工程、网站开发等许多专业领域。

相比于 C++、Java 等语言来说，Python 更加易于学习和掌握，并且可以利用其大量的内置函数与丰富的扩展库来快速实现许多复杂的功能。在 Python 语言的学习过程中，仍然需要通过不断地练习与体会来熟悉 Python 的编程模式，尽量不要将其他语言的编程风格用在 Python 中，而要从自然、简洁的角度出发，以免设计出低效率的 Python 程序。

本书的主要特色如下。

知识技术全面准确：本书主要针对国内计算机相关专业的高校学生以及程序设计爱好者们，详细介绍了 Python 语言的各种规则和规范，以便让读者能够全面掌握这门语言，从而设计出优秀的程序。

内容架构循序渐进：本书的知识脉络清晰明了，基础篇主要介绍 Python 的基本语法规则，提高篇主要讲解一些更加深层的概念，而应用篇则说明 Python 在具体应用场景中应当如何使用。本书内容由浅入深，便于读者理解和掌握。

代码实例丰富完整：对于书中每一个知识点都会配有一些示例代码并辅助相关说明文字及运行结果，还会在某些章节中对一些经典的程序设计问题进行深入的讲解和探讨。读者可以参考源程序上机操作，加深体会。

本书的配套教学课件及其他资源读者可登录华信教育资源网（www.hxedu.com.cn）注册后免费下载。本书中所有代码均能在 Python 2.7.11 中成功运行；对其稍加调整后也可以适用于 Python 3.x。

本书由吕云翔、赵天宇、张元编著，曾洪立、吕彼佳和姜彦华进行了素材整理和配套资源的制作。

由于 Python 的教学方法本身还在探索之中，加之编者的水平和能力有限，本书难免有疏漏之处，恳请各位同仁和广大读者批评指正，也希望各位能就实践过程中的经验和心得与编者进行交流（编者邮箱：yunxianglu@hotmail.com）。

编　者

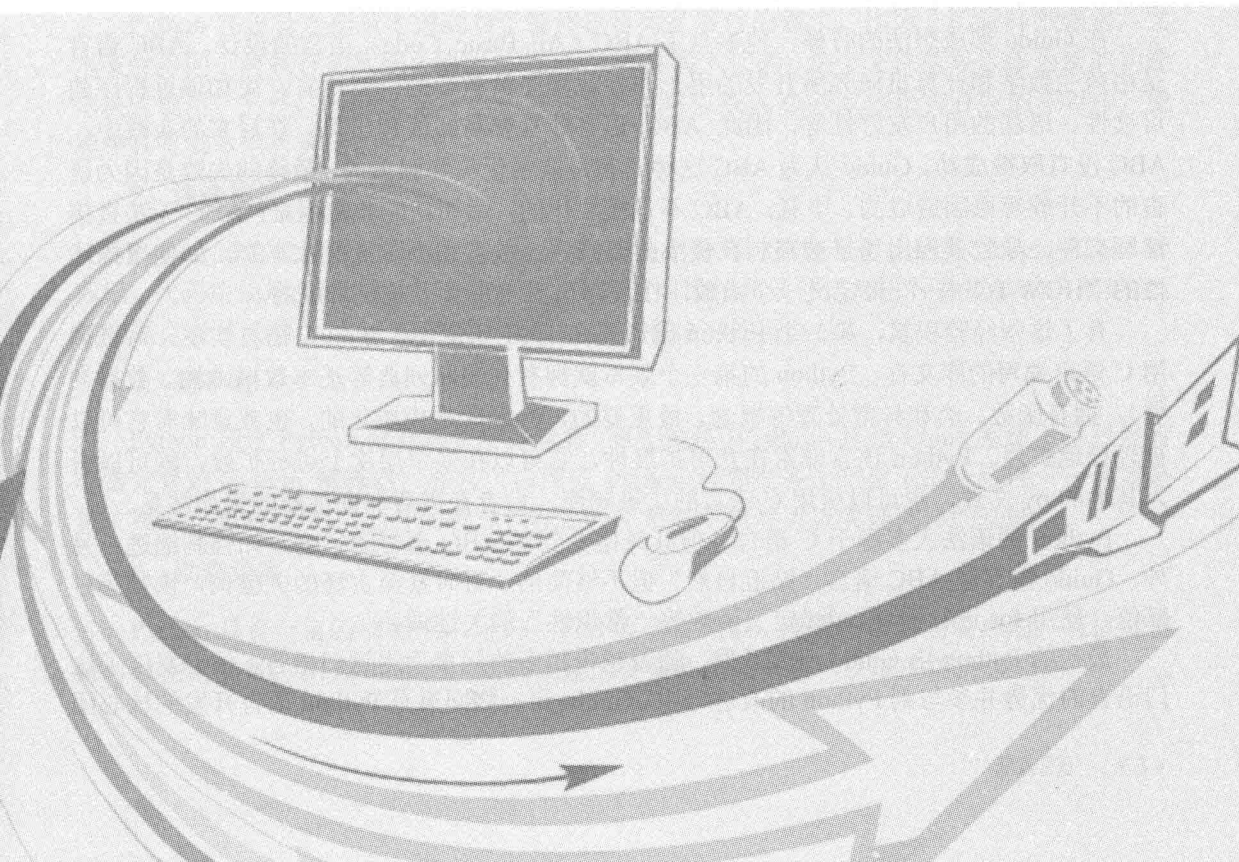
目　录

基　础　篇

进 阶 篇

应 用 篇

基　础　篇



第 1 章　Python 语言概述

1.1　Python 简史

Python 的英文直译是大蟒蛇，这个奇怪的名字有一个更加奇怪的出处——BBC 电视剧 Monty Python's Flying Circus——Python 语言设计者 Guido von Rossum 的最爱。

Guido 在 1982 年获得了阿姆斯特丹大学的数学和计算机硕士学位，在这两个专业中，他更倾向于与计算机。20 世纪 80 年代的计算机配置非常局限，与狭窄的现实相照应的，就是 Pascal、C、Fortran 等以“让机器运行得更快”为基本设计原理的许多高级编程语言的盛行。所有编译器的核心是代码优化，以防程序占满内存或占满 CPU。

Guido 使用过许多类似的语言，但在这些语言环境下的编程体验令他非常苦恼，因为程序员需要像计算机一样思考，为了迎合机器的口味和要求，榨取每一丝可能的存储空间和运算时间。这就导致，明明知道该如何通过某种语言写出一个功能，但实现的过程却要耗费大量的时间。Guido 看中了 UNIX 系统中的 Shell，他觉得程序员与系统通过 Shell 进行交互式操作的方法非常舒服，但是 Shell 本身不是一种编程语言，实现通用功能并不简单，也不能全面调用计算机的功能。于是他开始构思一种语言，它应具有 C 语言一般精确的问题解决能力，又可以像 Shell 一样轻松地编程。

在 Guido 形成想法的时候，他参与了 ABC（All Basic Code）语言的设计。ABC 语言是由荷兰数学和计算机研究所开发的用于非专业程序员教学的一门语言，更加偏重程序的可读性、语法的用户友好性等，因此 ABC 语言非常容易阅读和学习。在后来的实践中，ABC 没有取得成功。Guido 认为 ABC 这种语言是非常优美和强大的，最终的失败是因为语言的不开放等原因造成的。毕竟，ABC 不是模块化的，添加功能非常困难；ABC 无法直接读写文件，导致其应用场景被限制在极小的范围内；关键词不专业又太丰富，如使用描述性的“HOW TO……”来定义一个函数，在程序员看来事实上是反常规的。

有了这些经验积累，第一个 Python 编译器在 1991 年诞生，使用 C 语言实现，可以调用 C 语言编写的库文件。Python 的第一个版本就拥有列表和词典等基本数据结构（数据类型），拥有函数、类和异常处理等概念。最重要的，Python 是模块化的，也就意味着它可以被轻松地扩展。Python 语言非常在意可扩展性，它可以在多个层次上进行扩展，既可以引入其他的 py 文件，也可以引用 C 语言的二进制库，后者在需要保证性能时更加常见。

Python 的语法大多源于 C 语言，但其风格受到了 ABC 语言的影响，如强制缩进。另外，Guido 摒弃了 ABC 语言“贴近自然”但不够简洁、有时甚至古怪的关键词，恢复等号赋值、使用 for 进行循环、def 定义函数等“常识性”的关键词。

最初的 Python 由 Guido 一人开发，随着越来越多的同事使用这门语言，他们发现了这门语言的优势并参与到 Python 的改进当中。Guido 和一些同事是 Python 语言开发的核心团

队，使用业余时间来思考和完善语言特性。在研究所之外，Python 语言因为其得天独厚的可扩展性和语言自身的简洁性（隐藏了许多实现细节，更多地凸显逻辑过程）受到了广大程序员的青睐，Python 开始流行，用户越来越多，形成了强大的社区力量。扩展库也日益增多，功能随之强大。

在硬件性能不再拮据的今天，Python 作为解释型语言的劣势越来越小，它易于掌握，快速开发的优势在“大众编程”的互联网时代显得格外耀眼。据 2015 年 TIOBE 统计数据显示，Python 是全球流行度第八的编程语言。Python 语言以对象为核心组织代码，支持多种编程范式，采用动态类型并且自带内存回收机制。Python 可以解释运行，也可以调用 C 语言库进行扩展。Python 的标准库功能强大，而社区与组织不断提供的像 numpy、scipy、matplotlib、gevent、django 等优秀的第三方包也不断涌出，使得它成为低至编程新手，老到软件架构师都偏爱的一门编程语言。

1.2 Python 的语言特点

Python 是一门跨平台、开源、免费的解释型高级动态编程语言，通过一些工具可以进行伪编译，还可以将 Python 源程序转换为可执行程序。Python 支持命令式编程、函数式编程和面向对象编程，并且可以作为把多种不同语言编写的程序无缝衔接在一起的“胶水”语言，可以发挥出不同语言和工具的优势。Python 的语法简洁明了，保证了程序的可读性。Python 是模块化的语言，极易扩展，还有强大的社区力量支持，拥有大量的实用扩展库，安装和接入非常简单，大大提升了开发效率。

总结起来，Python 有以下几种特点。

1. 可扩展

Python 在设计之初就考虑到对于编程语言可扩展性的需求。作为一门解释型语言，文本文件等同于可执行的代码，创建一个 py 文件并写入代码，这个文件就可以作为新的功能模块来使用。另外，Python 支持 C 语言扩展，也可以嵌入 C 或 C++语言开发的项目中，使程序具有脚本语言灵活的特性。

2. 语法精简

Python 语言中涉及的关键字很少，不需要使用分号，也废弃了大括号、begin 和 end 等标记，代码块使用空格或制表符来分割，支持使用循环和条件语句进行数据结构的初始化。这些语言设计使得 Python 程序短小精悍，并且有很高的可读性。

3. 跨平台

Python 通过 Python 解释器来解释运行，而无论是 Windows 还是 Linux 系统下都已经有非常完善的 Python 解释器，并且可以保证 Python 程序在各个平台下的一致性——也就是说，在 Windows 下可以运行的程序，在 Linux 系统下仍然可以实现同样的功能。

4. 动态语言

Python 具有一定的动态性，与 JS、Perl 等语言类似，变量不需要明确声明，直接赋值就可以使用变量。在 Python 中，动态创建的变量的类型与第一次赋的值类型相同。

5. 面向对象

Python 语言具有很强的面向对象特性。面向对象编程，相比面向结构编程而言，大大降低了实际问题建模的复杂度。一方面，面向对象使程序设计与现实生活逻辑更加接近；另一方面，面向对象程序可以让各个组件分界更为明确，降低了程序的维护难度。面向对象的程序设计抽象出类和对象的属性和行为，将它们组织在一定作用域内，使用封装、继承、多态等方法来简化问题和明确设计。Python 在一定程度上简化了面向对象的具体实现，取消了保护类型、抽象类、接口等元素，将更多的控制权交给了程序员。

6. 具有丰富的数据结构

Python 内置的数据结构丰富而强大，包括元组、列表、字典、集合等。内置数据结构简化了程序设计，缩短了代码长度，并且符号简明易懂，方便使用和维护。

7. 健壮性

Python 提供了异常处理机制、堆栈跟踪机制和垃圾回收机制。异常处理机制可以捕获程序的异常并报错，堆栈跟踪对象能够找出程序出错的位置和原因，垃圾回收机制可以有效管理申请的内存区域，及时释放不需要的空间。

8. 强大的社区支持

Python 语言因其出色的品质，受到专业与业余编程人士的广泛推崇。许多爱好者和第三方组织也在积极地为 Python 提供实用库。目前，Python 语言正在 Web 开发、网络、图形图像、数学计算等领域大放异彩，是许多领域新手入门和项目开发的绝佳工具。

1.3 搭建 Python 开发环境

Python 开发环境的安装和配置非常简单。Python 可以在多个平台下进行安装和开发。本节将介绍如何下载并安装 Python 以及如何在命令行中使用 Python。

1.3.1 Python 的下载与安装

在搭建 Python 的开发环境之前，首先需要读者对 Python 的版本有一定的了解。官方提供了 Python 2.x 和 Python 3.x 两个版本，这两个版本彼此不兼容，其代码规范有一定区别，且很多内置函数的实现和使用方式也经过了修改，标准库也经过了重新整合。特别是，3.x 版本不是 2.x 版本的升级版，Python 官方仍然在对两个版本分别进行更新。由于扩展库的发行往往滞后于 Python 的升级，很多扩展库至今没有对 3.x 版本提供支持。因此，在安装开发环境之前，读者首先要了解自己使用 Python 的目的，再来选择适合自己的版本。在新的 Python 版本发布之后，也要在确定自己常用的扩展库对新版本进行了支持后再一起更新。

对不太了解 Python 的用户来说，版本选择有一个粗略的规则。Python 2.7 是一个“黄金版本”，绝大多数第三方库对此版本提供支持，所以如果需要大量辅助库的支持，通常 Python 2.7 是一个不错的选择。但如果读者还没有 Python 开发经验，甚至是第一次接触这

门程序语言，那么可放心地选择 Python 3.x 系列的最高版本。Python 3.x 版本带来了更规范的语法和更合理的标准库，有理由相信之后的一段时间第三方会普遍对 3.x 提供支持。

注意：

本书所有代码均在 Python 2.7 下运行通过。对于一些 Python 3.x 版本中与 Python 2.x 不同的特性，本书会进行简单说明。

Python 的各个版本可以在官方网站 https://www.python.org/downloads 获取，选择相应版本会进入下载信息页面，如图 1-1 所示。

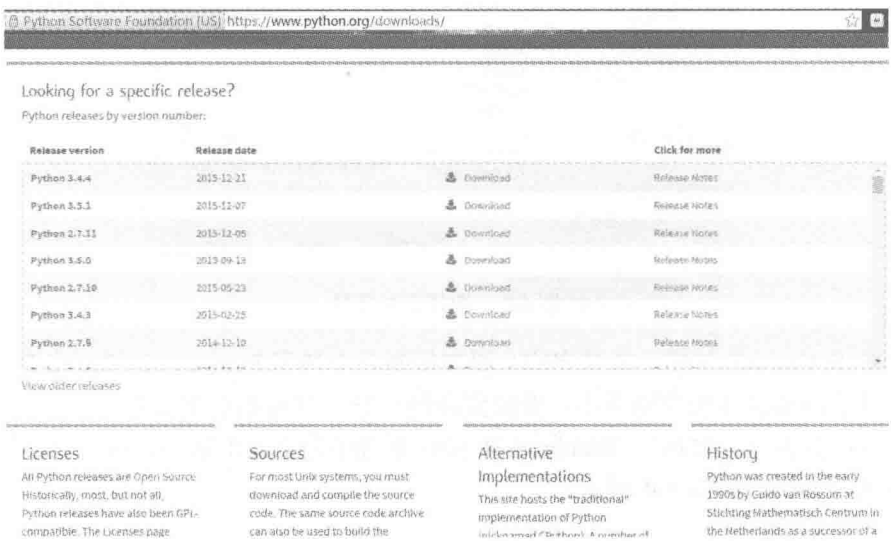


图 1-1　选择合适的 Python 版本

单击一个目标版本（如 Python 2.7.11），会进入如图 1-2 所示的下载页面，页面下方的表格中提供了各个目标操作系统对应的下载项。本书若无特殊说明，均在 Windows 10 的 64 位系统环境下运行，所以这里选择“Windows x86-64 MSI installer”选项。此时会启动安装文件的下载。

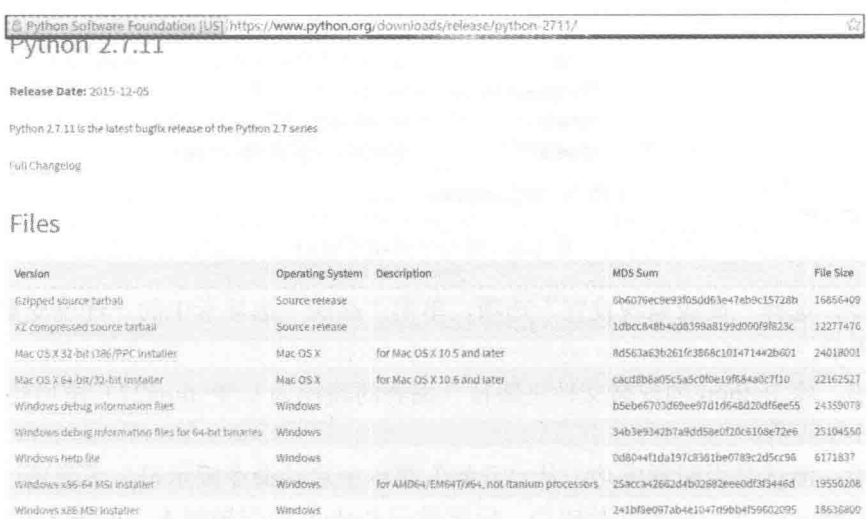


图 1-2　选择相应的操作系统

待下载结束后，双击 msi 文件进行安装，选择安装路径，如图 1-3 所示。

图 1-3　选择安装路径

继续单击“Next”按钮，安装完毕！要在控制台中使用 Python，还需要将 Python 的解释器所在目录添加到环境变量当中。要配置环境变量，可遵循以下步骤。

第一步：右击“计算机”（Windows 10 操作系统中为“此电脑”），选择“属性”选项，打开“系统”窗口，如图 1-4 所示。

图 1-4　进入“系统”窗口

第二步：选择“高级系统设置”选项，单击“高级”选项卡中的“环境变量”按钮，如图 1-5 所示。

第三步：将 Python 的安装文件夹路径（编者的路径为 C:\Python27）添加到 Path 变量中，如图 1-6 所示。

第四步：测试是否配置成功，从“开始”菜单中启动命令提示符，或者通过运行启动 cmd，输入 python –V。如果安装成功，应该显示 Python 的安装版本，如图 1-7 所示。