中国高校
创意创新创业
教育系列丛书

王学颖 刘立群 刘冰 司雨昌 编著

Python学习从入门到实践

清华大学出版社

中国高校创意创新创业教育系列丛书

王学颖 刘立群 刘冰 司雨昌 编著

Python
学习从入门到实践

清华大学出版社
北京

内 容 简 介

本书是一本适合 Python 初学者学习程序设计与开发的基础教程，从应用的角度介绍了 Python 的发展、基本语句与语法、数据与运算、程序基本结构、函数与模块、面向对象和文件处理。本书既注重知识的系统性，又兼顾了内容的实用性，既保持了结构的严谨完整，又体现了语言的清晰简洁。

本书设置了丰富的教学案例，帮助读者用最简单直观的方式理解知识。同时，本书选取了 Python 常用的第三方库函数的应用实例，内容涉及图形绘制、中文分词、图形用户界面、网络爬虫、数据库访问等，引导读者进行深入的学习和研究。

本书内容具有知识完整、通俗易懂、叙述简练的特点，适合各层次读者使用，既可以作为高校计算机课程的教材，也可以供初学者或专业人士阅读。本书配套的电子资源包括 PPT、案例代码、习题等，均提供免费下载。

图书在版编目(CIP)数据

Python 学习：从入门到实践/王学颖等编著．—北京：清华大学出版社，2017(2018.2重印)
(中国高校创意创新创业教育系列丛书)
ISBN 978-7-302-48697-8

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

中国版本图书馆 CIP 数据核字(2017)第 270409 号

责任编辑：谢　琛
封面设计：常雪影
责任校对：焦丽丽
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954
印 装 者：三河市君旺印务有限公司
经　　销：全国新华书店
开　　本：210mm×235mm　　**印　　张**：17.75　　**字　　数**：367 千字
版　　次：2017 年 12 月第 1 版　　**印　　次**：2018 年 2 月第 2 次印刷
印　　数：2000~4000
定　　价：48.00 元

产品编号：075598-01

前　言

Python 语言是一种面向对象的解释型计算机程序设计语言，它既支持面向过程的编程，也支持面向对象的编程。Python 的语法简洁，没有过多的语法细节要求，其代码可读性强且更高效。Python 具有优秀的可拓展性，至今已有 11 万余个标准库和第三方库，可以方便地实现顶层和底层的黏性扩展，被称为胶水语言。Python 语言是一种完全开源的语言，因此被广泛使用，据 TIOBE 编程语言排行榜统计，截至 2017 年 5 月，Python 语言位于编程语言排行榜第四，仅次于 Java、C、C++ 语言。

“高级语言程序设计基础”是高校普遍开设的一门计算机基础课程，它面向计算机专业和非计算机专业的学生，主要目标是通过程序设计语言的学习，使学生掌握程序设计的基本思想和方法，培养和训练分析解决问题的思维习惯。Python 语言以其优美、清晰、简单的语法特点，非常适合作为第一门程序设计语言，它不仅非常容易掌握，更重要的是，Python 语言利用其丰富的函数库可以方便地开发面向各学科领域的应用，是学生进行专业学习和研究的有力工具。可以说，Python 是一种“一学就会”并使人终身受益的程序设计语言。

本书就是在上述背景下编写的，读者对象主要是编程零基础的学生。书中内容强调通俗易懂、简洁清晰、由浅入深。全书共分为 10 章，主要内容包括 Python 语言概述、数据类型和表达式、控制语句、数据结构、字符串和正则表达式、函数和模块、类和对象、文件处理、异常处理以及高级应用。

本书内容覆盖了 Python 的全部知识点，并且对每一个重要知识都设置了程序设计实例，强化对核心知识点的解读，引导学生通过具体案例掌握程序设计的方法。在案例的选择上，本书注重趣味性和实用性，使实例贴近生活、面向专业，既改变了程序设计的刻板生硬，又具有一定的实际应用价值。

本书由王学颖、刘立群、刘冰、司雨昌共同编著，在编写过程中参考了许多任课教师的意见和建议，在此向这些老师表示衷心的感谢。

本书提供了丰富的教学资源，内容包括教学 PPT、教学案例、习题和答案。

本书在写作过程中参考了大量的书籍和资料，在此向这些文献的作者表示衷心的感谢。

由于作者水平有限，书中难免有不足之处，敬请广大读者提出宝贵意见。

作　者

2017 年 5 月

目　录

第 1 章

Python 语言概述

学习目标

- 了解计算机语言的演变
- 了解高级语言的运行机制
- 掌握 Python 语言环境的安装与运行
- 掌握 Python 语言的基本语法

1.1 从计算机到编程

1.1.1 程序语言的演变

从第一台计算机宣告问世以来，程序就成为计算机的代名词。正如计算机所经历的 70 年飞速发展一样，编程也已经远远不是当初的样子。

编程其实就是把人类的需求用计算机语言表达，是人与计算机的对话。计算机语言(Computer Language)是人与计算机之间通信的语言，是人与计算机之间传递信息的媒介。计算机语言经历了从机器语言、汇编语言到高级语言的演变过程。

机器语言是用二进制代码表示的计算机能直接识别和执行的一种指令的集合，它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言是计算机唯一能够识别的语句，由 0 和 1 构成指令的集合。这样的机器语言十分复杂、不易阅读和修改，而且容易产生错误。程序员们很快就发现了使用机器语言带来的麻烦，它们难以辨别和记忆，给整个产业的发展带来了障碍，于是，汇编语言产生了，如图 1.1 所示。

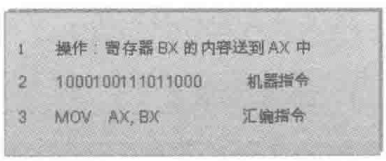

图 1.1 从机器指令到汇编指令

汇编语言是经过符号化的计算机语言，它用一些简洁的英文字母和符号替代二进制指令，相对于枯燥的机器代码而言更易于读写、调试和修改。汇编语言需要一个专门的程序将这些符号翻译成相应的二进制机器指令，这种翻译程序称为汇编程序。汇编语言直接面向处理器，它与硬件关系密切，因此通用性和可移植性较差。也正因为如此，它的程序执行代码短、执行速度快、效率较高，因此在一些时效性要求较高的程序以及许多大型程序的核心模块和工业控制方面得到大量应用。

高级语言主要是相对于汇编语言而言的，是一种比较接近自然语言和数学公式的编程语言。高级语言用人们更易理解的方式编写程序，基本脱离了机器的硬件系统，有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且易于学习和掌握。计算机语言的演变见表 1.1。

表 1.1 计算机语言的演变

计算机语言	编写方式及要素	特　　点
机器语言	二进制编码 操作码、地址码	速度快，效率高，占用内存少 直观性差，难以纠错，编写需要很强的专业性
汇编语言	助记符号 操作码、地址码	速度快，效率高，占用内存少，直观性较强 编写专业性较强
高级语言	接近自然语言的语法 源程序、编译或解释程序	占用内存多，执行需要编译 易于掌握，可读性强 独立性、共享性及通用性强

例如，在高级语言中表示“将 BX 的内容送到 AX”，使用的代码为 AX=BX。

高级语言编写的程序称为高级语言源程序，但是高级语言并不是特指的某一种具体的语言，而是包括很多种编程语言，如流行的 Java、C、C++ 、C＃、Pascal、Python 语言等，这些语言的语法以及命令格式都不尽相同。

1.1.2 高级语言的运行机制

高级语言按照执行方式可以分为编译型和解释型两种。

1. 编译型语言

编译型高级语言是通过专门的编译器，将高级语言代码（源程序）一次性翻译成可执行的机器码（目标程序）的编程语言，这个翻译过程称为 Compile。编译生成的可执行程序可以脱离开发环境，在特定的平台上独立运行。常用的编译型语言有 C、C++、FORTRAN 等。

编译程序对源程序进行解释的方法相当于日常生活中的整文翻译。在编译程序的执行过程中，要对源程序扫描一遍或几遍，最终形成一个可在具体计算机上执行的目标程序。通过编译后可以产生高效运行的目标程序，目标程序可以不依赖于程序环境而被多次执行。

编译型语言具有如下优点。

(1) 可独立运行，源代码经过编译形成的目标程序可脱离开发环境独立运行。

(2) 运行效率高，编译过程包含程序的优化过程，编译的机器码运行效率较高。

2. 解释型语言

解释型语言是通过解释器对高级语言代码(源程序)逐行翻译成机器码并执行的语言。每次执行程序都要进行一次翻译，因此解释型语言的程序运行效率较低，不能脱离解释器独立运行。常用的解释型语言有 Basic、Python 等。

解释程序对源程序进行翻译的方法相当于日常生活中的同声传译。解释程序对源程序的语句从头到尾逐句扫描、逐句翻译，并且翻译一句执行一句，因而这种翻译方式并不形成机器语言形式的目标程序。

解释型语言的优点如下。

(1) 易于修改和测试，逐句解释过程中便于对代码进行修改和测试。

(2) 可移植性较好，只要有解释环境，就可在不同的操作系统上运行。

1.2 Python 的产生与特性

1.2.1 Python 语言的发展

Python 语言的作者 Guido von Rossum 是荷兰人。1982 年，Guido 在阿姆斯特丹大学(University of Amsterdam)获得了数学和计算机硕士学位。在那个时候，他接触并使用过诸如 Pascal、C、FORTRAN 等语言，这些语言的基本设计原则是让机器能更快地运行。在 20 世纪 80 年代，个人计算机的配置很低，如早期的 Macintosh 只有 8MHz 的 CPU 主频和 128KB 的 RAM，一个大的数组就能占满内存，所有编译器的核心是进行优化，以便让程序能够运行。为了提高效率，程序员需要像计算机一样思考，以便能写出更符合机器口味的程序。这种编程方式让 Guido 感到苦恼，他知道如何用 C 语言写出一个功能，但整个编写过程需要耗费大量的时间。Guido 受到 UNIX 系统的解释器 Bourne Shell 的启发，Python 应运而生。

Python 这个名字来自 Guido 所挚爱的电视剧 *Monty Python's Flying Circus*，他希望这个叫做 Python 的新语言，能符合他的理想：创造一种 C 和 Shell 之间的功能全面、易

学易用、可拓展的语言。

1991 年，第一个 Python 编译器(同时也是解释器)诞生了。它是用 C 语言实现的，并能够调用 C 库，已经具有了类(class)、函数(function)、异常处理(exception)，包括了列表(list)和词典(dictionary)等数据类型，以及以模块(module)为基础的拓展系统。

由于 Python 的开源性，使它经历了一个快速发展的阶段。

```
Python 2.0-2001/06/22
Python 2.4-2004/11/30
Python 2.5-2006/09/19
Python 2.6-2008/10/01
Python 2.7-2010/07/03
Python 3.0-2008/12/03
Python 3.1-2009/06/27
Python 3.2-2011/02/20
Python 3.3-2012/09/29
Python 3.4-2014/03/16
Python 3.5-2015/09/13
Python 3.6-2016/12/23
```

Python 3.0 相对早期的版本是一个较大的升级，但是 Python 3 在设计时没有考虑向下兼容，所以很多早期版本的 Python 程序无法在 Python 3 上运行。为了照顾早期的版本，Python 推出了过渡版本 2.6，它基本使用了 Python 2.x 的语法和库，同时考虑了向 Python 3.0 的迁移，允许使用部分 Python 3.0 的语法与函数。2010 年继续推出了兼容版本 2.7。2014 年 11 月，Python 发布消息，将在 2020 年停止对 2.0+版本的支持，并且不会再发布 2.8 版本，建议用户尽可能地迁移到 3.4+版本。

1.2.2 Python 语言的特性

Python 崇尚优美、清晰、简单，是一个优秀并广泛使用的语言。

1. 语法简单

Python 语法很多来自 C 语言，但又与 C 语言有很大的不同。Python 程序没有太多的语法细节和规则要求，采用强制缩进的方式。这些语法规定让代码的可读性更好，编写的代码质量更高，程序员能够更简单、高效地解决问题，使编程能够专注于解决问题而不是语言本身。

2. 可移植性

用 Python 编写的代码可以移植在许多平台上，这些平台包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE，甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台等。

3. 黏性扩展

Python 又被称为胶水语言，它具有优秀的可拓展性。Python 可以在多个层次上拓展，既可以在高层引入 py 文件，也可以在底层引用 C 语言的库。Python 已经拥有 11 万余个标准库和第三方库，它可以完成的操作包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、GUI(图形用户界面)等操作。因此，Python 的编程就像钢结构房屋一样，程序员可以在此框架下自由地任意搭建功能。

4. 开源理念

Python 语言是一种开源语言，使用者可以自由地发布这个软件的复制、阅读它的源代码、对它进行改动、把它的一部分用于新的自由软件中等。正是由于 Python 的完全开源，Python 吸引了越来越多优秀的人加入进来，形成了庞大的 Python 社区。如今，各种社区提供了成千上万的开源函数模块，而且还在不断地发展。

5. 面向对象

Python 既支持面向过程的函数编程，也支持面向对象的抽象编程。在面向过程的语言中，程序是由过程或可重用代码的函数构建起来的，而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 以一种非常强大而简单的方式实现面向对象编程。

1.3　Python 的安装与运行

1.3.1　Python 的下载和安装

搭建 Python 的开发环境，就是指安装 Python 的解释器。IDLE 是开发 Python 程序的基本 IDE(集成开发环境)，具备基本的 IDE 功能，是非商业 Python 开发的不错选择。只要安装 Python 以后，IDLE 就自动安装好了。

Python的安装非常简单。首先，需要进入Python官方网站http://www.python.org/downloads/下载适合操作系统的安装包，如图1.2所示。

图1.2 Python官方网站

网页页面会显示发布的最新版本，会随着陆续的发布随时进行更新。在这里要根据相应的操作系统选择不同的版本，如选择Python For Windows，如图1.3所示。在列出的版本列表中选择一个版本，单击即可下载。

Python releases by version number:

Release version	Release date	
Python 3.6.1	2017-03-21	Download
Python 3.4.6	2017-01-17	Download
Python 3.5.3	2017-01-17	Download
Python 3.6.0	2016-12-23	Download
Python 2.7.13	2016-12-17	Download
Python 3.4.5	2016-06-27	Download
Python 3.5.2	2016-06-27	Download

图1.3 Python官网发布的更新

Python 3对Python 2进行了重大升级，Python已经不再进行Python 2的后续更新，Python 3不完全向下兼容2.x程序。本书所有程序和案例全部基于Python 3，建议初学者选择3.x版本进行下载并安装，如图1.4所示。