“十三五”普通高等教育规划教材

Python程序设计基础教程

吕云翔　等编著

提供电子课件、动画视频

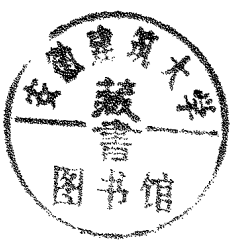
http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

十三五普通高等教育规划教材

Python 程序设计基础教程

吕云翔　孟　爻　赵天宇　张　元　郭若冲　编著



机械工业出版社

Python 是一门简单易学、功能强大的编程语言，拥有高效的高层数据结构，特别适用于快速应用程序开发。本书共分为 16 章，主要内容包括：Python 简介、Python 环境搭建、函数、模块、文件操作、字符串与正则表达式、面向对象编程、异常处理、Python 基本概念、Python 控制结构、Python 多线程与多进程编程、使用 Python 进行 GUI 开发、使用 Python 进行数据管理、Python Socket 网络编程、使用 Python 进行 Web 开发，以及 Python 综合应用实例。

本书既可以作为高等院校相关专业的教材，也可以作为程序设计爱好者的学习指导用书。

本书配套授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取。QQ：2850823885。电话：010-88379739。

图书在版编目（CIP）数据

Python 程序设计基础教程 / 吕云翔等编著. —北京：机械工业出版社，2018.5
“十三五”普通高等教育规划教材
ISBN 978-7-111-60316-0

Ⅰ. ①P… Ⅱ. ①吕… 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（2018）第 141416 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：郝建伟 责任编辑：郝建伟
责任校对：张艳霞 责任印制：孙炜
天津千鹤文化传播有限公司印刷
天津千鹤文化传播有限公司装订
2018 年 8 月第 1 版 • 第 1 次印刷
184mm×260mm • 13.75 印张 • 332 千字
0001－3000 册
标准书号：ISBN 978-7-111-60316-0
定价：45.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：（010）88379833
读者购书热线：（010）88379649

网络服务
机 工 官 网：www.cmpbook.com
机 工 官 博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
金 书 网：www.golden-book.com

前　言

Python 是一门解释型、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 20 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，也被广泛应用到统计分析、计算可视化、图像工程和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，仍然需要通过不断的练习与体会来熟悉 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，以免设计出冗长且低效的 Python 程序。

本书的主要特色如下。

知识技术全面准确：本书主要针对国内高校相关专业的学生及程序设计爱好者，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，第 1～5 章主要介绍 Python 的基本语法规则，第 6～9 章主要讲解一些更加深层的概念，而第 10～16 章则选取了 Python 在一些当下流行的具体应用场景下的使用方法。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：针对书中的每一个知识点都会配有一些示例代码，并辅以相关说明文字及运行结果，某些章节还会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。

微课辅助学习：在某些章节，尤其是有关实际编程的章节，辅助有视频讲解。

本书中所有的代码均能在 Python 2.7.11 版本下成功运行，对其稍加调整后也可以适用于 Python 3.x 版本。

本书由吕云翔、孟爻、赵天宇、张元、郭若冲编著。

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，本书难免存在疏漏之处。恳请各位同仁和广大读者给予批评指正，也希望各位读者能将实践过程中的经验和心得与我们交流（yunxianglu@hotmail.com）。

编　者

目　录

第 1 章　Python 简介

本章主要介绍 Python 语言的起源、发展及推广的过程；然后简单明了地阐述 Python 语言的特点，以及广受开发者好评的原因。

1.1　Python 的发展历程

自从 20 世纪 90 年代初 Python 语言诞生至今，它已被逐渐广泛应用于系统管理任务的处理和 Web 编程。

Python 的创始人为 Guido van Rossum。1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。之所以选中 Python（中文意为“大蟒蛇”）作为该编程语言的名称，是因为他是一个名为 Monty Python 的喜剧团体的爱好者。

ABC 是由 Guido 参与设计的一种教学语言。就 Guido 本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有取得成功，究其原因，Guido 认为是其非开放造成的。Guido 决心在开发 Python 的过程中避免这一错误。同时，他还想实现在 ABC 中闪现过但未曾实现的东西。

就这样，Python 在 Guido 的手中诞生了。可以说，Python 是从 ABC 发展起来的，主要受到了 Modula-3（另一种相当优美且强大的语言，为小型团体所设计的）的影响，并且结合了 UNIX Shell 和 C 的习惯。

1991 年，第一个 Python 编译器（也是解释器）诞生。它是用 C 语言实现的，并能够调用 C 语言的库文件。自诞生之日起，Python 就已经具有类、函数、异常处理、包含表和词典在内的核心数据类型，以及模块为基础的拓展系统。

Python 语法很多来自 C，但又受到 ABC 语言的强烈影响。来自 ABC 语言的一些规定直到今天还富有争议，如强制缩进。但这些语法规定让 Python 容易读。另一方面，Python 聪明地选择服从一些惯例，特别是 C 语言的惯例，如回归等号赋值。Guido 认为，如果是“常识”上确立的东西，没有必要过度纠结。

Python 从一开始就特别在意可拓展性。Python 可以在多个层次上拓展。在高层，可以直接引入.py 文件；在底层，可以引用 C 语言的库。Python 程序员可以快速地使用 Python 编写.py 文件作为拓展模块。但当性能是考虑的重要因素时，Python 程序员可以深入底层，编写 C 程序，编译为.so 文件引入到 Python 中使用。Python 就好像是使用钢构建房一样，先规定好大的框架，而程序员可以在此框架下相当自由地拓展或更改。

最初的 Python 完全由 Guido 本人开发。Python 得到了 Guido 的同事们的欢迎。他们迅速地反馈使用意见，并参与到 Python 的改进中来。Guido 和一些同事组成 Python 的核心团队。他们将自己大部分的业余时间都用于研究 Python。随后，Python 拓展到研究所之外。

Python 将许多机器层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思考。Python 程序员可以花更多的时间用于思考程序的逻辑，而不是具体的实现细节。这一特征吸引了广大程序员，Python 开始流行。

Python 被称为 Battery Included，是说它标准库的功能强大。这是整个社区的贡献。Python 的开发者来自不同领域，他们将不同领域的优点带给 Python。比如 Python 标准库中的正则表达是参考 Perl，而 lambda、map、filter 和 reduce 等函数参考了 Lisp。Python 本身的一些功能及大部分的标准库来自于社区。Python 的社区不断扩大，进而拥有了自己的 newsgroup、网站及基金。从 Python 2.0 开始，Python 也从 maillist 的开发方式转为完全开源的开发方式。社区氛围已经形成，开发工作被整个社区分担，Python 也获得了更加高速的发展。

到今天，Python 的框架已经确立。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，自动进行内存回收。Python 支持解释运行，并能调用 C 库进行拓展。Python 拥有强大的标准库。由于标准库的体系已经稳定，所以 Python 的生态系统开始拓展到第三方包，如 Django、web.py、wxpython、numpy、matplotlib 和 PIL 等。

2017 年 7 月，根据 IEEE Specturm 研究报告显示，在 2016 年排名第三的 Python 已经成为世界上最受欢迎的语言，C 和 Java 分别位居第二和第三位。

1.2 Python 的语言特点

Python 是一种面向对象、直译式计算机程序设计语言，这种语言的语法简捷而清晰，具有丰富和强大的类库，基本上能胜任用户平时需要的编程工作。

可以写一个 UNIX Shell 脚本或 Windows 批处理文件完成任务，然而 Shell 脚本更擅长于移动文件和修改文本数据，不适合于图形界面应用程序或游戏。可以编写一个 C、C++ 或 Java 的程序，但是就算一个简单的方案草案，也需要花费大量的时间。Python 更易于使用，可在 Windows、Mac OS X 和 UNIX 操作系统上使用，并会帮助用户更快速地完成工作。

Python 简单易用，但它是一个真正的编程语言，比 Shell 脚本或批处理文件提供了更多的结构和对大型程序的支持。另一方面，Python 比起 C 提供了更多的错误检查，同时作为一门高级语言，它具有高级的内置数据类型，如灵活的数组和字典。由于 Python 提供了更为通用的数据类型，比起 Awk 甚至 Perl，它适合更宽广的问题领域。同样在做许多其他的事情上，Python 也不会比别的编程语言更复杂。

Python 允许用户将自己的程序分成不同的模块，可以在其他 Python 程序中重用这些模块。它配备了一个标准模块，可以自由使用这些标准模块作为程序的基本结构，或者作为例子开始学习 Python 编程。这些模块提供了类似文件 I/O、系统调用、网络编程，甚至像 Tk 的用户图形界面工具包。

Python 是一门解释性语言，它可以在程序开发期节省相当多的时间，因为它不需要编译和链接。Python 解释器可以交互使用，这使得用户很容易体验 Python 语言的特性，以便于编写发布用的程序，或者进行自下而上的开发。它也是一个方便的桌面计算器。

Python 让程序可以写得很健壮和具有可读性，用 Python 编写的程序通常比 C、C++或 Java 要短得多，其原因如下。

1）高级的数据类型使用户在一个语句中可以表达出复杂的操作。

2）语句的组织是通过缩进而不是开始和结束括号。

3）不需要变量或参数的声明。

Python 是可扩展的：如果用 C 编写程序就很容易为解释器添加一个新的内置函数或模块，也能以最快速度执行关键操作，或者使 Python 程序能够连接到所需的二进制架构上（如某个专用的商业图形库）。一旦真正选择了 Python，可以将 Python 解释器连接到用 C 编写的应用上，使得解释器作为这个应用的扩展或命令性语言。

由于 Python 语言的简洁性、易读性及可扩展性，在国外用 Python 做科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。例如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。众多开源的科学计算软件包都提供了 Python 的调用接口，如著名的计算机视觉库 OpenCV、三维可视化库 VTK 和医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了，例如下列 3 个十分经典的科学计算扩展库：NumPy、SciPy 和 matplotlib，它们分别为 Python 提供了快速数组处理、数值运算及绘图功能。因此 Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。

习题

一、简述题

1．查阅资料，了解 Python 在不同方向的应用，以及对应的库有哪些。

2．查阅资料，了解 Guido 发明 Python 的趣事。

第 2 章　Python 环境搭建

本章主要介绍 Python 在主流平台（如 Windows、Linux、UNIX、Mac）上的安装、环境搭建等；然后用经典的“Hello，Python”带领读者首次创建属于自己的 Python 程序，对 Python 这门语言有一个直观的认识。

2.1　Python 安装

在开始编程之前，需要安装一些新软件。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，可以直接访问 http://www.python.org/download，下载并安装 Python 的最新版本。

Python 在不同平台下的安装方式不同，本节分为 Windows、UNIX&Linux 和 Mac 共 3 种平台进行介绍。

2.1.1　在 Windows 平台上安装 Python

在 Windows 平台上安装 Python 的方法如下。

1）打开 Web 浏览器访问网站http://www.python.org/download/。

2）在下载列表中选择 Windows 平台安装包，包格式为 python-XYZ.msi 文件，XYZ 为要安装的版本号。

3）要使用安装程序 python-XYZ.msi, Windows 系统必须支持 Microsoft Installer 2.0 搭配使用。只要保存安装文件到本地计算机，然后运行它，看看机器是否支持 MSI。Windows XP 和更高版本已经有 MSI，很多老机器也可以安装 MSI。

4）下载后，双击下载包，进入 Python 安装向导，安装非常简单，只需使用默认的设置一直单击“下一步”按钮直到安装完成即可。

2.1.2　在 UNIX & Linux 平台上安装 Python

目前很多 Linux 发行版如 Ubuntu 等已经预装了 Python 2.7 或 Python 3 的环境，如果没有预装，可以按照以下方法安装。

1）打开 Web 浏览器访问网站http://www.python.org/download/。

2）选择适用于 UNIX/Linux 的源码压缩包。

3）下载并解压压缩包。

4）如果需要自定义一些选项，则修改 Modules/Setup。

5）执行 ./configure 脚本。

6）执行 make 命令。

7）执行 make install 命令。

8）执行以上操作后，Python 会安装在/usr/local/bin 目录中，Python 库安装在/usr/local/lib/pythonXX 目录中，XX 为所使用的 Python 的版本号。

2.1.3 在 Mac 平台上安装 Python

最近的 Mac 系统都自带有 Python 环境，也可以在网站 http://www.python.org/download/ 上下载最新版安装。

2.2 Windows 下的环境变量配置

以下为在 Windows 10 中配置 Python 环境变量的具体步骤。

首先找到 Python 的安装位置，如图 2-1 所示（默认安装至 C 盘）。

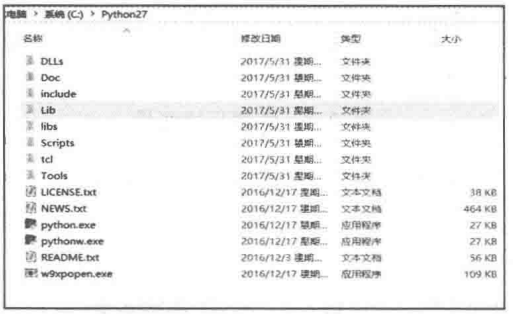


图 2-1 默认 Python 安装路径

复制 Python 的路径，右击“计算机”图标，在弹出的快捷菜单中选择“属性”命令，然后进入“高级系统设置”中，单击“环境变量”按钮，弹出“环境变量”对话框，如图 2-2 所示。

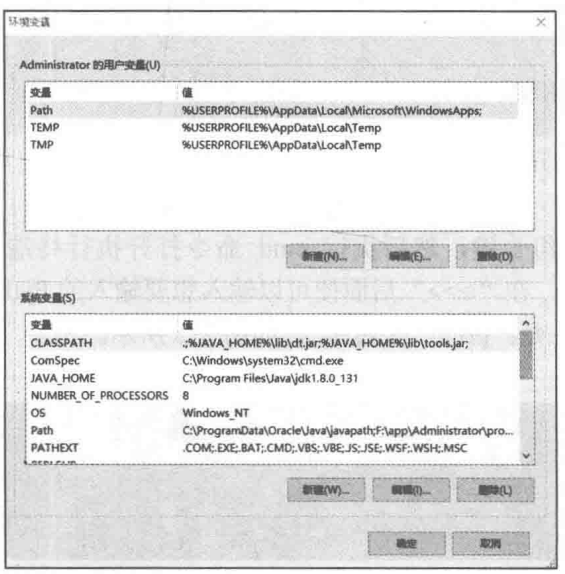


图 2-2 “环境变量”对话框

在“系统变量”中找到path，向其中添加Python路径，如图2-3所示。

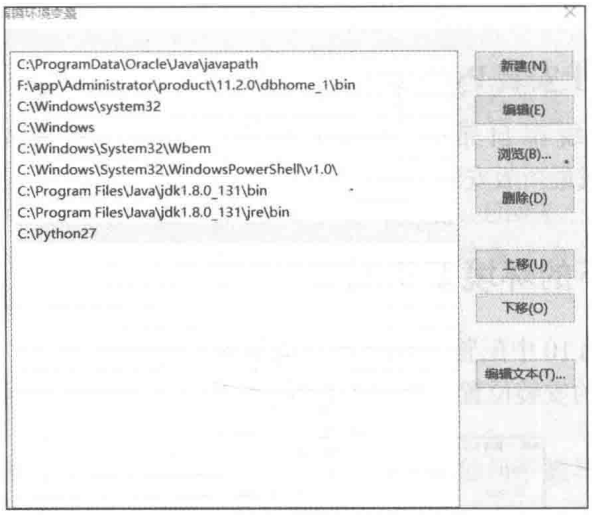


图2-3　编辑环境变量

然后检验Python是否已安装好，进入命令行，输入python，若得到以下信息则表示已经安装好了，如图2-4所示。

图2-4　验证Python安装与配置

2.3 Hello, Python

Python脚本应用的开发有两种方式，一种是进入Python的交互环境下开发，另一种则是直接编写脚本文件。

按〈Windows+R〉组合键，然后执行cmd命令打开执行终端，输入python，当出现“>>>”则说明成功进入，在“>>>”后面便可以输入想要输入的Python语句，举例如下。

1）print的输出方法是：print‘字符串’，如图2-5所示。

图2-5　使用print输出字符串

按〈Enter〉键执行后，就可以看到输出内容了。

2）退出交互环境的函数：exit()，如图 2-6 所示。

图 2-6　使用 exit()函数退出交互环境

可以看到已经退出 Python 的交互环境了。但是这种方法显得很麻烦，每次还得手工输入命令，并且该命令符窗口关闭以后，前面所做的操作全部都会无效。所以，在实际开发时还是以新建脚本文件（Python 的脚本文件以.py 结尾），然后编写该脚本，最后执行该脚本的流程学习使用，具体步骤如下。

1）新建一个 test.py 文件。

2）用文本方式打开该文件（若在 Window 环境下建议用 notepad++文本编辑器）。

3）输入：print 'Hello,Python'。

4）保存。

5）在同目录下新建一个 cmd.bat 文件，输入 cmd.exe 保存（该方式是快速进入工作目录）。

6）输入 python test.py 查看执行效果，会发现 'Hello,Python' 会被输出出来，如图 2-7 所示。

图 2-7　使用 Python 脚本文件

习题

一、简述题

查阅资料，了解 Windows 环境变量中 path 的含义或 Linux 中/bin 目录的特点。

二、实践题

在 Windows 或 Linux 系统中自己搭建 Python 环境，完成“Hello、World”的输出。

第3章　Python基本概念

本章以介绍 Python 这门语言的基础概念为主，首先讲述 Python 这门语言的基本数据类型；然后介绍运算符，包括算术运算符、关系运算符、逻辑运算符、位运算符、身份运算符和成员运算符等；从而引出表达式、常用模块及函数的概念，并对基本输入/输出给出了明确的解释。

3.1　基本数据类型

计算机可以处理各种各样的数据，不同的数据需要定义不同的数据类型来存储。数据类型决定了如何将代表这些数据值的位存储到计算机的内存中。例如，整数 25 和字符串 Python 会在计算机内存中用不同的方式来存储和组织。

Python 的基本数据类型包括整型、浮点型、字符串、布尔值和空值等。

3.1.1　整型

Python 可以处理任意大小的整数，也包括负整数。十进制整数的表示方式与数学上的写法相同，如 10、-255、0、2016 等。此外，Python 还支持十六进制、八进制和二进制整数。

十六进制整数需要用 0x 或 0X 作为前缀，用 0～9 和 a～f 作为基本的 16 个数字来表示，如 0xffff、0x4f5da2 等。

八进制整数需要用 0 作为前缀，用 0～7 作为基本的 8 个数字来表示，如 011、0376 等。

二进制整数需要用 0b 或 0B 作为前缀，用 0 和 1 作为基本数字来表示，如 0b1010，0b10110 等。

代码清单 3-1 演示了 Python 中的几种不同进制下的整数及长整数的使用方法。

代码清单 3-1

```
1    print 2016
2    print 0xffff
3    print 0376
4    print 0b101101
```

【输出结果】

```
2016
65535
254
44
```

实际上，Python 中的整数可以分为普通整数和长整数。普通整数对应 C 语言中的 long

类型，其精度至少为 32 位；长整数具有无限的精度范围。当所创建的整数大小超过普通整数取值范围时将自动创建为长整数，也可以通过对数字添加后缀 L 或 l 来手动创建一个长整数。

3.1.2 浮点型

在 Python 中，浮点型用来表示实数，绝大多数情况下用来表示小数。浮点数可以采用普通的数学写法，如 1.234、-3.14159、12.0 等。

对于特别大或特别小的浮点数，可以使用科学计数法表示，如-1.23e11、3.2e-12 等。其中，使用字母 e 或 E 来表示 10 的幂。因此上面的两个例子就表示-1.23×10^{11}和3.2×10^{-12}。

3.1.3 复数

除整数和浮点数外，Python 还提供了复数作为其内置类型之一，如 3+2j、7-2j 等。其中 j 代表虚数单位。

3.1.4 字符串

字符串是使用单引号或双引号括起来的任意文本，如'Hello World'，"Python"等。请注意引号本身不是字符串的一部分，只说明了字符串的范围。例如，字符串'ab'只包含 a 和 b 两个字符。使用 "或" "可以表示空字符串。

一个字符串使用哪种引号开头就必须以哪种引号结束。例如字符串"I'm"就包含了 I、' 和 m 共 3 个字符，字符串的结束是双引号而非单引号。

通过以上说明，可以知道字符串'He's good'是不合法的，因为字符串将在第二个单引号处结束，后边的字符部分和第三个单引号成为非法部分。针对这个问题有两种解决方法，第一种方法是将外部的引号换成双引号，将字符串变为"He's good"。但当字符串中包含了两种引号时这种方法就无效了。

第二种方法是使用转义字符（\）来标识出引号。通过在某些字符前加上转义字符可以表示特别的含义。在上面所说的情况下，通过在引号前加上反斜杠来打印引号。因此，上述字符串可以被写作'He\'s good'。同样，\"用来在字符串中表示一个双引号字符。

除了对引号进行转义，转义字符还用来表示一些特殊的字符。例如，\n 表示换行符，即一行的结束。Python 中常用的转义字符如表 3-1 所示。

表 3-1 Python 中的转义字符

转义字符	名称	ASCII 值
\b	退格符	8
\t	制表符	9
\n	换行符	10
\f	换页符	12
\r	回车符	13
\\	反斜线	92
\'	单引号	39
\"	双引号	34