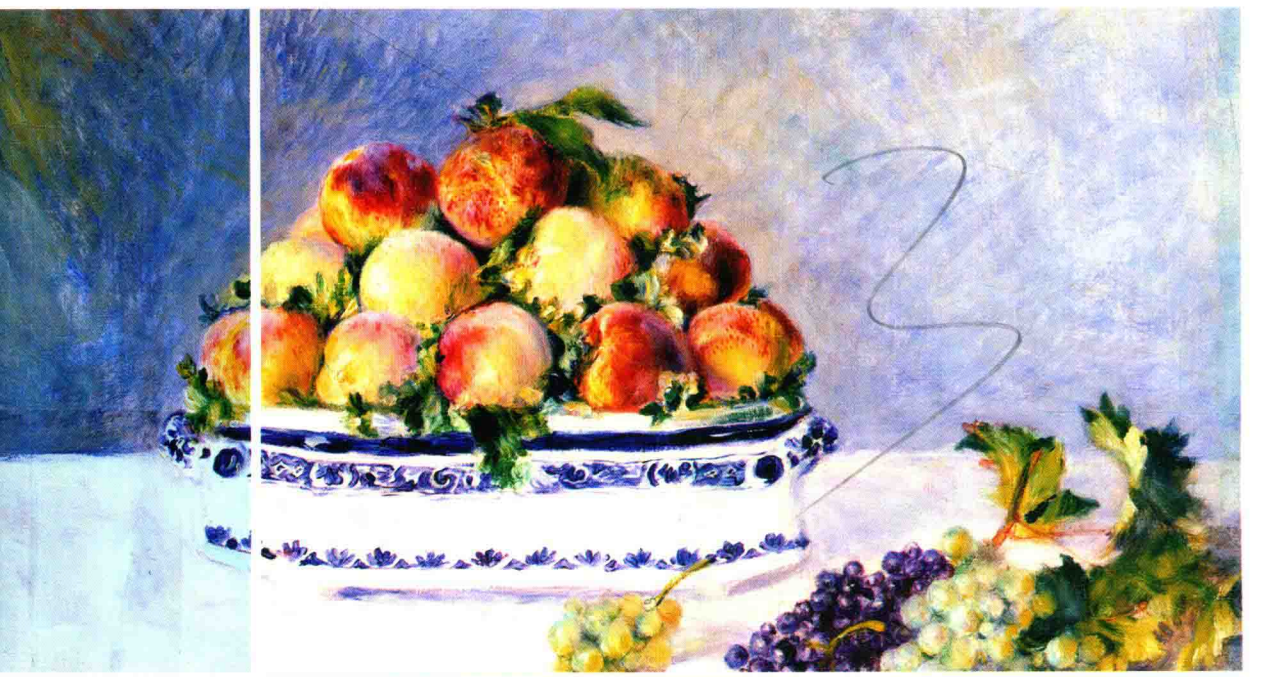
21世纪高等学校计算机
基础实用规划教材

Python程序设计入门

◎ 吕云翔 孟爻 编著



零基础入门·案例式教学·微课视频讲解·教学资源

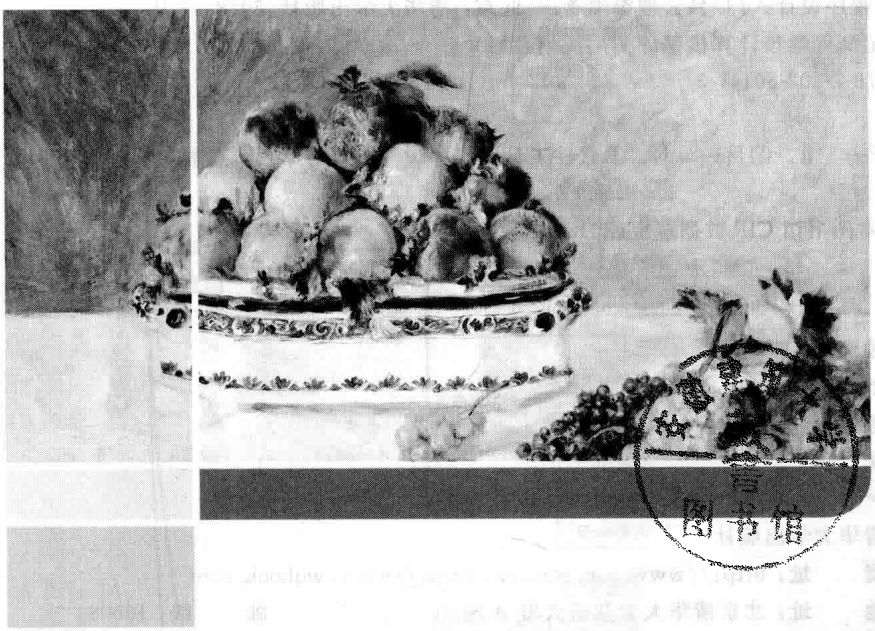
200分钟
视频讲解

清华大学出版社

21世纪高等学校计算机
基础实用规划教材

Python程序设计入门

◎ 吕云翔 孟爻 编著



清華大學出版社
北京

内 容 简 介

Python是一种简单易学，功能强大的编程语言，它有高效率的高层数据结构，特别适用于快速的应用程序开发。全书共分为15章，主要内容包括Python简介、Python环境搭建、Python基础语法、函数、模块、文件操作、异常处理、面向对象编程、正则表达式、Python GUI编程、Python多线程与多进程编程、Python访问数据库、Python Socket网络编程、Python Web编程以及Python综合应用实例。

本书既可以作为普通高校计算机相关专业的教材，也可以作为Python爱好者的参考书。

图书在版编目(CIP)数据

Python程序设计入门/吕云翔等编著. —北京：清华大学出版社，2018
(21世纪高等学校计算机基础实用规划教材)
ISBN 978-7-302-50147-3

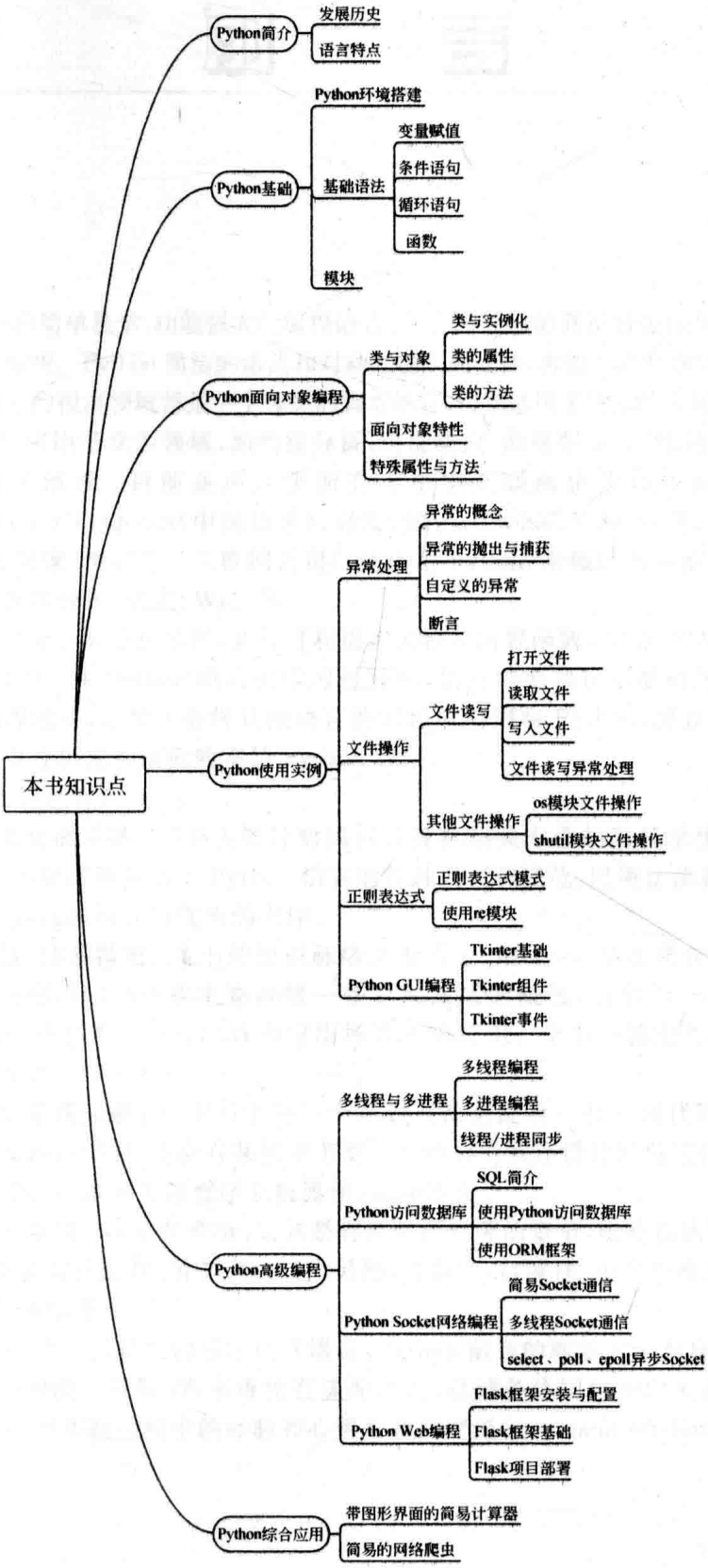
Ⅰ. ①P… Ⅱ. ①吕… Ⅲ. ①软件工具－程序设计－高等学校－教材 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2018)第112365号

责任编辑：魏江江　薛　阳
封面设计：刘　键
责任校对：徐俊伟
责任印制：沈　露

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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：14.5　　**字　　数**：356千字
版　　次：2018年8月第1版　　**印　　次**：2018年8月第1次印刷
印　　数：1～1500
定　　价：39.00元

产品编号：074945-01


本书知识点
Python简介
发展历史
语言特点
Python基础
Python环境搭建
基础语法
变量赋值
条件语句
循环语句
函数
模块
Python面向对象编程
类与对象
类与实例化
类的属性
类的方法
面向对象特性
特殊属性与方法
Python使用实例
异常处理
异常的概念
异常的抛出与捕获
自定义的异常
断言
文件操作
文件读写
打开文件
读取文件
写入文件
文件读写异常处理
其他文件操作
os模块文件操作
shutil模块文件操作
正则表达式
正则表达式模式
使用re模块
Python GUI编程
Tkinter基础
Tkinter组件
Tkinter事件
Python高级编程
多线程与多进程
多线程编程
多进程编程
线程/进程同步
Python访问数据库
SQL简介
使用Python访问数据库
使用ORM框架
Python Socket网络编程
简易Socket通信
多线程Socket通信
select、poll、epoll异步Socket
Python Web编程
Flask框架安装与配置
Flask框架基础
Flask项目部署
Python综合应用
带图形界面的简易计算器
简易的网络爬虫

前　言

Python是一种简单易学，功能强大的编程语言，它有高效率的高层数据结构，能简单而有效地实现面向对象编程。Python简洁的语法和对动态输入的支持，再加上解释型语言的本质，使得它在大多数平台上的很多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发。

Python可以应用于众多领域，如数据分析、组件集成、网络服务、图像处理、数值计算和科学计算等众多领域。目前业内几乎所有大中型互联网企业都在使用Python，如Youtube、Dropbox、BT、Quora(中国知乎)、豆瓣、知乎、Google、Yahoo!、Facebook、NASA、百度、腾讯、汽车之家、美团等。互联网公司广泛使用Python来做的事一般有自动化运维、自动化测试、大数据分析、爬虫、Web等。

Python更加易于学习和掌握，并且可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，而要从自然、简洁的角度出发，以免设计出冗长而低效率的Python程序。

本书的主要特色有：

- **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，书中详细介绍了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。
- **内容先进、体系得当**：本书的知识脉络清晰明了，第1～4章主要介绍Python的基本语法规则，第5～9章主要讲解一些更加深层的概念，而第10～15章则选取了Python一些在当下流行的具体应用场景下的应用。全书内容由浅入深，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以相关说明文字及运行结果，还会有某些章节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，配有视频讲解。

本书的编著者为吕云翔、孟爻、徐祺智，另外，曾洪立、吕彼佳、姜彦华参与了部分章节的编写及配套资源制作等。

由于Python是一门新兴的程序设计语言，Python语言的教学方法本身还在探索之中，加之我们的水平和能力有限，本书难免有疏漏之处，恳请各位同仁和广大读者给予批评指正，也希望各位能将实践过程中的经验和心得与我们交流(yunxianglu@hotmail.com)。

编　者

2018年5月

目　录

第1章 Python简介

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作为该编程语言的名字，是因为他是一个叫Monty Python的喜剧团体的爱好者。

ABC是由Guido参加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语言并没有成功，究其原因，Guido认为是其非开放造成的。Guido决心在Python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手中诞生了。可以说，Python是从ABC发展起来，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。从一出生，Python已经具有了类、函数、异常处理，包含表和词典在内的核心数据类型，及以模块为基础的拓展系统。

Python语法很多来自C语言，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，例如强制缩进。但这些语法规定让Python容易读。另外，Python聪明地选择服从一些惯例，特别是C语言的惯例，例如回归等号赋值。Guido认为，如果“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。从高层上，用户可以直接引入.py文件。在底层，用户可以引用C语言的库。Python程序员可以快速地使用Python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，写C程序，编译为.so文件引入到Python中使用。Python就好像是使用钢结构建房一样，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Guido同事的欢迎。他们迅速地反馈使用意见，并参与到Python的改进工作中。Guido和一些同事构成Python的核心团队。他们将自己大部分的业余时间用于研究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的程序员，Python 开始流行。

Python 被称为“Battery Included”，是说它以及其标准库的功能强大。这些是整个社区的贡献。Python 的开发者来自不同领域，他们将不同领域的优点带给 Python。例如 Python 标准库中的正则表达是参考 Perl，而 lambda、map、filter、reduce 等函数参考了 Lisp。Python 本身的一些功能以及大部分的标准库来自社区。Python 的社区不断扩大，进而拥有了自己的 newsgroup、网站，以及基金。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。社区气氛已经形成，工作被整个社区分担，Python 也获得了更加高速的发展。

到了今天，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 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。这些包，如 Django、web. py、wxpython、numpy、matplotlib、PIL，将 Python 升级成了“物种”丰富的“热带雨林”。

Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为 2010 年度语言。自从 2004 年以后，Python 的使用率呈线性增长。

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，这种语言的语法简洁而清晰，具有丰富和强大的类库，基本上能胜任我们平时需要的编程工作。

我们可以写一个 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，而不适合图形界面应用程序或游戏。我们可以写一个 C/C++/Java 的程序，但就算是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，但它是一种真正的编程语言，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。另外，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，例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。同时，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，用户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 Tkinter 的用户图形界面工具包。

Python 是一种解释型语言，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 解释器可以交互地使用，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：如果用户知道用 C 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 程序能够链接到所需的二进制架构上(例如某个专用的商业图形库)。一旦真正迷上了 Python，可以将 Python 解释器连接到用 C 写的应用上，使得解释器作为这个应用的扩展或命令行语言。

由于 Python 语言的简洁性、易读性以及可扩展性，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，例如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了，例如以下三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、数值运算以及绘图功能。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 1

一、选择题

1. Python 在(　　)年的圣诞期间被荷兰人 Guido van Rossum 发明。

A. 1988　　B. 1989　　C. 1990　　D. 1999

2. 第一个 Python 解释器于(　　)年问世。

A. 1989　　B. 1990　　C. 1991　　D. 1999

3. Python 从(　　)版本开始完全开源。

A. 1.5　　B. 2.0　　C. 2.7　　D. 3.0

二、填空题

1. Python 是一种_____、_____式的计算机程序设计语言，这种语言的语法简洁而清晰，具有丰富和强大的类库，基本上能胜任我们平时需要的编程工作。

2. Python 使用_____来组织语句、代码块。

3. Python _____(是/否)需要声明变量或参数。

三、论述题

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

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，我们分为 Windows、UNIX&Linux、MAC 三种平台分别介绍。

2.1.1 Windows 安装 Python

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

(2) 在下载列表中选择 Window 平台安装包，包格式为：python-XYZ.msi 文件，XYZ 为我们要安装的版本号。

(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，看看我们的机器是否支持 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，只需要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 UNIX & Linux 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，可以按照如下方法安装。

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

(2) 选择适用于 UNIX/Linux 的源码压缩包。

(3) 下载及解压压缩包。

(4) 如果需要自定义一些选项修改 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，XX 为我们使用的 Python 的版本号。

2.1.3 MAC 安装 Python

最近的 MAC 系统都自带有 Python 环境，我们也可以在链接 http://www.python.org/download/上下载最新版进行安装。

2.2 Windows 下环境变量的配置

以下为 Windows 10 中配置 Python 环境变量的具体步骤：

(1) 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 路径

(2) 复制 Python 的路径，右击计算机图标，选择“属性”，然后进入高级系统设置中，单击环境变量，如图 2.2 所示。

(3) 在系统变量中找到 path，向其中添加 Python 路径，如图 2.3 所示。

(4) 检验 Python 是否装好，进入命令行，输入“python”，得到如下信息表示已经安装好，如图 2.4 所示。

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。

使用 Python 交互环境开发的步骤如下。

同时按下 Win 键+R 键，然后执行 cmd 打开执行终端，输入 Python 命令，当出现：“>>>”说明成功进入，在>>>后面便可以输入想要输入的 Python 语句，例如：

(1) print 的输出方法是：print “字符串”，如图 2.5 所示。

按 Enter 键执行后，我们就可以看到内容输出了。

(2) 退出交互环境的函数：exit()，如图 2.6 所示。

图 2.2　环境变量界面

图 2.3　编辑环境变量

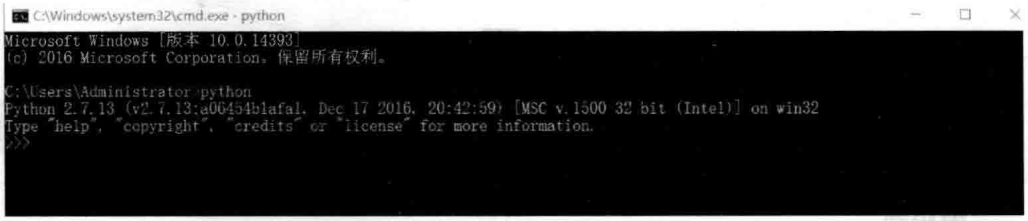


图 2.4　验证 Python 安装与配置

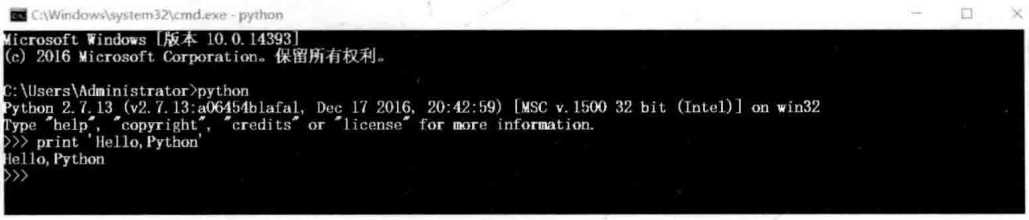


图 2.5　使用 print 输出字符串

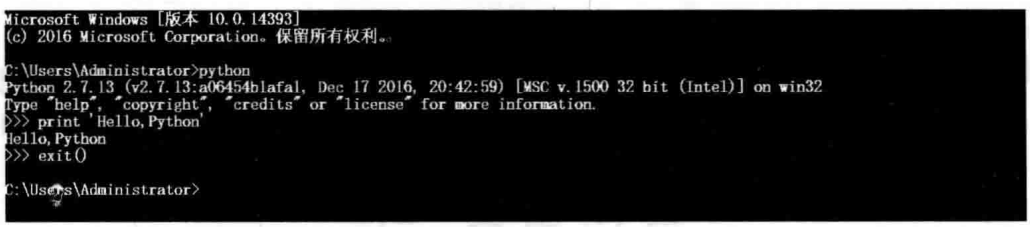


图 2.6　使用 exit()函数退出环境

可以看到我们已经退出 Python 的交互环境了。但是这种方法显得很麻烦，每次需人为输入命令，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，我们在实际开发的时候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) 文本方式打开该文件(Windows 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'被输出，如图 2.7 所示。

图 2.7　使用 Python 脚本文件