

Python

编程与应用实践

主编 冯伟森 王 建 杨 宁

Python BIANCHENG
YU YINGYONG SHIJIAN



四川大学出版社

Python

编程与应用实践

主编 冯伟森 王 建 杨 宁

Python BIANCHENG
YU YINGYONG SHIJIAN



四川大学出版社

责任编辑:毕 潜
责任校对:蒋 玑
封面设计:墨创文化
责任印制:王 炜

图书在版编目(CIP)数据

Python 编程与应用实践 / 冯伟森, 王建, 杨宁主编.
—成都: 四川大学出版社, 2019. 3
ISBN 978-7-5690-2835-5

I. ①P… II. ①冯… ②王… ③杨… III. ①软件工
具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2019) 第 056249 号

书名 Python 编程与应用实践

主 编 冯伟森 王 建 杨 宁
出 版 四川大学出版社
地 址 成都市一环路南一段 24 号 (610065)
发 行 四川大学出版社
书 号 ISBN 978-7-5690-2835-5
印 刷 郫县犀浦印刷厂
成品尺寸 185 mm×260 mm
印 张 18
字 数 460 千字
版 次 2019 年 3 月第 1 版
印 次 2019 年 3 月第 1 次印刷
定 价 68.00 元



版权所有◆侵权必究

- ◆读者邮购本书, 请与本社发行科联系。
电话: (028)85408408/(028)85401670/
(028)85408023 邮政编码: 610065
- ◆本社图书如有印装质量问题, 请
寄回出版社调换。
- ◆网址: <http://press.scu.edu.cn>

前 言

Python 语言是一门流行的开源编程语言，广泛应用于各个领域的独立程序与脚本化应用中。它不仅免费、可移植、功能强大、相对简单，而且使用起来充满乐趣，在人工智能、大数据分析、科学计算、大型网站服务搭建等领域发挥着不可替代的重要作用。特别是在以大数据和机器学习为应用背景的各种项目开发中，Python 语言已经是开发的首选语言，可以使用 Python 快速地编写人脸识别、机器翻译、语音合成、机器人、网络爬虫等各种有趣的应用软件。作者结合多年编程教学经验编写此书，力求通过通俗易懂的讲授、丰富的教学资源、便捷的互助工具，使编程学习变得简单有趣。

本书共分三个部分，第一部分主要对基础数据类型、字符串、序列、程序结构、函数、模块、类、文件等面向过程的内容进行了通俗易懂的讲解。第二部分对表、栈、树、图等常见数据结构进行了介绍，同时对查找、排序、文本处理等典型的实用算法进行了深入的阐述。第三部分通过一个有趣的游戏案例对所有内容进行了综合应用，使学习者能对 Python 的知识点有更深入的理解和实际的应用体验。全书内容丰富，条理清晰，层次分明，逻辑性强，内容和方法前后贯通，阐述深入浅出，并配有大量的难易程度不同的练习题，可作为非计算机专业研究生、本科、专科程序设计课程教材，也可作为计算机专业本科、专科程序设计基础课程教材，还可作为 Python 爱好者自学用书。根据我们的经验，使用本书可在 80 学时内完成全部教学任务。

内容	章节	建议学时数
Python 基础知识	1、2、3、4、5、6、7、8、15	48
数据结构	9、10、11	24
数据结构与算法分析	12、13、14	16

本书的最大特点是为解决初学者在学习 Python 编程时所容易遇到的问题，配套提供了在线编程智能学习平台 (<http://www.dingdangcode.com>)，该平台提供了三大工具：一是提供了教学课件、视频、案例的动画演示、讲义、习题等丰富的教学资源，供用户更好地理解书中的内容；二是提供了在互联网中可以使用的在线编程环境，用户无须复杂和烦琐的安装和环境配置，直接打开浏览器即可编写代码并看到效果；三是用户在使用中如

果有任何编程问题无法解决时，可以通过平台在线提问，平台中会有许多爱好编程的伙伴为你提供准确的解答思路和参考答案，帮助学习者快速地入门并喜欢上 Python 编程。

本书的第 1~9 章由冯伟森编写，第 10~11 章由王建编写，第 12~14 章由杨宁编写，第 15 章由王建编写，全书由冯伟森统稿。在本书的编写过程中，我们参阅了大量的相关书籍和资料，在此向有关作者表示衷心的感谢。四川大学计算机学院的研究生赵丽娅、武蒙蒙、霍君慧、伍艺、万珂蓝、蒋仕艺、张文涛等参与了本书的编写，并对书中的全部代码进行了调试，在此向他们表示诚挚的感谢。

编 者

2019 年 2 月

目 录

第一部分 Python 基础知识

第 1 章 Python 概述	(3)
1.1 什么是 Python	(3)
1.2 Python 的开发环境	(7)
1.3 Python 的应用	(13)
第 2 章 基本数据类型	(16)
2.1 变量和赋值	(16)
2.2 数字	(16)
2.3 字符串	(20)
习 题	(22)
第 3 章 序列	(24)
3.1 列表	(24)
3.2 元组	(25)
3.3 字典	(26)
3.4 集合	(28)
习 题	(32)
第 4 章 程序结构	(34)
4.1 条件语句	(34)
4.2 循环语句	(36)
4.3 continue 和 break 语句	(40)
4.4 迭代器	(41)
习 题	(42)
第 5 章 函数与模块	(45)
5.1 函数的定义	(45)
5.2 函数创建和调用	(45)
5.3 参数	(46)
5.4 返回值	(49)
5.5 命名空间和作用域	(50)
5.6 生成器	(51)
5.7 模块和 import 语句	(55)

5.8 包	(56)
5.9 Python 标准库	(56)
习 题	(57)
第 6 章 面向对象	(60)
6.1 什么是对象	(60)
6.2 类	(61)
6.3 继承	(62)
6.4 方法	(64)
6.5 组合	(68)
6.6 子类和派生	(69)
6.7 私有化	(70)
习 题	(71)
第 7 章 字符编码与文件 i/o	(73)
7.1 字符编码	(73)
7.2 文件对象	(73)
7.3 二进制文件操作	(74)
习 题	(78)
第 8 章 异常处理	(79)
8.1 什么是异常	(79)
8.2 检测和处理异常	(79)
8.3 raise 语句和 with 语句	(81)
习 题	(82)

第二部分 数据结构与算法分析

* 第 9 章 线性结构	(90)
9.1 线性表	(90)
9.2 栈	(101)
9.3 队列	(105)
习 题	(110)
* 第 10 章 树	(114)
10.1 树的基本概念	(114)
10.2 二叉树	(116)
10.3 二叉树的遍历	(119)
10.4 哈夫曼树与哈夫曼编码	(126)
习 题	(135)
* 第 11 章 图	(137)
11.1 图的定义和术语	(137)
11.2 图的存储表示	(139)
11.3 图的遍历	(143)

11.4 图的最小生成树.....	(144)
11.5 有向无环图的应用.....	(149)
11.6 最短路径问题.....	(152)
习 题.....	(157)
* * 第 12 章 查找	(159)
12.1 查找的基本概念.....	(159)
12.2 静态表的查找.....	(159)
12.3 动态查找.....	(163)
12.4 散列表.....	(186)
习 题.....	(190)
* * 第 13 章 排序	(191)
13.1 排序概述.....	(191)
13.2 插入排序.....	(192)
13.3 交换排序.....	(196)
13.4 选择排序.....	(204)
13.5 归并排序.....	(212)
13.6 基数排序.....	(215)
习 题.....	(218)
* * 第 14 章 文本处理	(220)
14.1 模式匹配.....	(220)
14.2 文本压缩.....	(228)
习 题.....	(233)

第三部分 Python 项目实战

第 15 章 飞机大战	(237)
15.1 项目介绍与准备.....	(237)
15.2 开始项目.....	(238)
15.3 项目总结.....	(253)
参考文献.....	(254)
附录 I 叮当码智能学习平台使用指南.....	(256)
附录 II 叮当码智能教学平台（高校版）使用指南.....	(266)

第一部分 Python 基础知识

本书的第一部分主要介绍 Python 的基础知识，这些知识很多都适用于多门编程语言，因此它们将在编程学习中发挥很大的作用。

第 1 章主要介绍了 Python 是什么，如 Python 的发展历史、特点以及 Python 与其他编程语言的关系和对比。同时也介绍了 Python 的开发环境以及环境的配置，还有 Python 在这些环境下的具体使用示例。最后介绍了 Python 的一些实际应用场景，以及 Python 的未来发展方向。

第 2 章主要介绍了 Python 中什么是变量，怎么给变量赋值，以及几种常见的数据类型，这些基本的数据类型将会是我们今后编写程序代码的重要组成部分。

第 3 章主要论述了更复杂的 Python 组合方式（列表、元组、字典、集合），以及怎么用它们来解决更复杂的实际问题。

第 4 章介绍了 Python 中的程序结构，主要是对 if 条件语句和循环语句进行了介绍，Python 中也支持 for 循环和 while 循环。本章还介绍了一些辅助语句，如 continue 和 break，以及迭代器的使用。

第 5 章介绍了 Python 中的函数和模块，主要是关于函数的定义、创建、调用以及如何设置参数等方面，讨论了一些高级特性，如生成器的创建、使用、生成等，论述了 Python 中模块和包的概念以及如何在项目中对它们进行导入。函数是执行特定任务的被命名的代码块，用户可以根据需要随时运行它。

第 6 章主要介绍了什么是面向对象的编程，进一步学习面向对象中类、方法、继承、组合等相关知识，以便于在今后可以进行结构式、过程式的开发。

第 7 章主要介绍了字符编码与文件 i/o，常见的字符编码以及什么是文件对象，对二进制文件的读写操作也进行了详细介绍并举例说明。

第 8 章主要介绍了异常的概念，以及如何检测和处理异常。

第 1 章 Python 概述

本章首先介绍了 Python 是什么，如 Python 的发展历史、特点以及 Python 与其他编程语言的关系和对比。其次介绍了 Python 的开发环境以及环境的配置，Python 在这些环境下的具体使用示例。最后介绍了 Python 的一些实际应用场合，以及 Python 的未来发展方向。

1.1 什么是 Python

本节从 Python 的发展历史、设计定位、特点，以及 Python 与其他语言的关系和对比等方面进行了介绍。

1.1.1 Python 的发展历史

Python 由荷兰人 Guido van Rossum（如图 1-1 所示）发明于 1989 年，第一个公开发行版发行于 1991 年。在 1989 年的圣诞节期间，Guido 为了打发在阿姆斯特丹无聊的时间，决心开发一个新的脚本解释程序，而 Python 这个名字，则是取自 Guido 挚爱的英国六人喜剧团体——巨蟒剧团（Monty Python）。

Python 的诞生，其实也伴随着一定的时代背景。在 20 世纪 80 年代，由于个人电脑的配置低，可供存储和计算的空间十分有限，所以程序员们必须绞尽脑汁最大化利用空间。这让 Guido 感到苦恼，因为这样编写程序太耗费时间，而且效果也不一定好，所以他想到了 shell——一种调用命令的程序设计语言，用 C 需要写上百行的程序在 shell 下只用几行就能实现，但其缺点是并不能全面调动计算机的功能。如果有一种语言能结合 C 和 shell 的优点是最好不过的了。后来，Guido 进入了 CWI（Centrum Wiskunde & Informatica）工作，并参与了 ABC 语言的开发。ABC 语言的初衷就是为了使程序变得易读、易懂、易学习，但其缺点也十分显著，首先就是需要配置较高的电脑才能运行，所以可扩展性差，而且也不能直接进行 I/O 操作等。而 Python 就是在 ABC 语言的基础上做了优化和扩展而最终成形的。



图 1-1 Guido van Rossum

Python 改进了 ABC 语言的缺陷，并且结合了 C 与 shell 的优点，既能全面调用计算机功能接口，又能快速轻松地编写程序，简单易学，可读性强。Python 的第一个编译器是用 C 语言实现的，故从 Python 一诞生，就能调用 C 库（.so 文件）。不仅如此，Python 还包含了类（Class）、函数（Function）、异常处理（Exception），核心数据类型除了常见的以外，还有列表（List）和字典（Dictionary），以及以模块（Module）为基础的扩展系统。Python 沿用了大部分 C 语言的语法习惯，并且注意了可扩展性，再加上它出现在个人计算机开始普及的年代，因此得到了一定程度的传播。

Python 易于扩展和改造的特点，使得 Python 集中了不同领域的优点。刚开始的时候，对于 Python 的不同需求，用户可以将改动发给 Guido，Guido 最后来决定是否将改动加入 Python 或其标准库中。随着 Python 的不断发展，到 Python 2.0 时，已经是完全开源的开发方式了，故 Python 又再次得到了更为广泛的传播。

Python 发展至今，其框架已经确立，有了稳定的标准库体系。现在的 Python 功能比以前更加强大。Python 以对象为核心来组织代码，比起 C 的面向过程，面向对象使得系统更灵活、更易于维护。Python 还支持各种编程范式（命令式编程、声明式编程、函数式编程等），能自动进行内存回收，支持解释运行等。Python 有强大的标准库作为支撑，并且随着各种第三方扩展包运用于其上，使得它越来越多地用于独立的、大型的项目开发中。图 1-2 为 2002 年至 2018 年 TIOBE 编程语言世界排行榜。

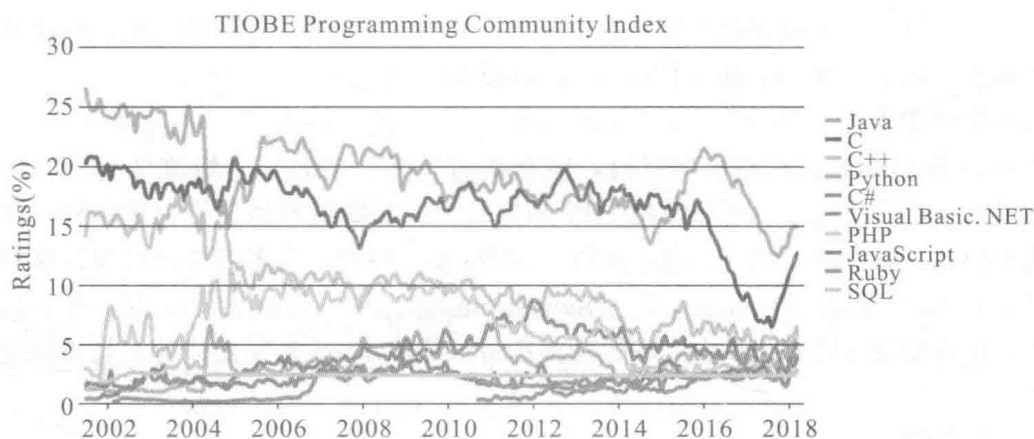


图 1-2 TIOBE 编程语言世界排行榜

1.1.2 Python 的特点

Python 的设计理念是“优雅”“明确”“简单”。Python 是一种面向对象的编程语言，延续了 C 语言能调用计算机各种功能接口的优点，以及 C 语言编程上书写的一部分习惯，同时又实现了 shell 一类的“胶水语言”的作用，使得编程方便简洁且快速有效，最后还改进了 ABC 语言电脑配置要求高以及扩展性低的缺点，使得 Python 拥有了强大的生命力。作为一款功能强大的面向对象编程语言，Python 主要具有以下特点：

(1) 明确简单。

在设计 Python 语言时，开发者就尽量选择了明确的、歧义较少的语法，故 Python 的可读性很强。由于明确简单、结构清晰，极大地减少了学习成本，所以初学者能快速上手操作。因为不用过于担心程序编写时在形式上的诸多细节问题，所以总体而言 Python 的编程是较为自由和简单的。在形式上注意的问题少意味着可以专注程序本身的逻辑，事实上也确实如此，因为编写规则简单，使用 Python 时更能将注意力集中在程序本身的逻辑和算法之上。

(2) 面向对象。

Python 是一种面向对象的编程语言。Python 自诞生开始就拥有了“类”的概念。对象包括了函数、模块、数字等，而“类”的封装就是面向对象最直接的体现。Python 还支持继承（包括多重继承）、重载和派生。面向对象程序设计的优点也同样在 Python 中得到体现：因为封装从而形成了许多“模块”，这样的机制不仅提高了程序的可读性，而且在每次维护的时候都不用进行全局维护，局部维护相较于全局维护更加方便和低成本；继承、重载、派生这些概念使得程序代码的可重用性和可扩展性得到提升，在开发时运用面向对象的抽象化思想则能提高效率和质量；封装也使得程序编写时是专注于局部效果实现的，所以在用 Python 语言编写程序时无须考虑硬件和内存等底层问题，这也使得 Python 更加简单有效。

(3) 开源免费。

Python 是自由/开源软件（FLOSS）之一。开源即开放源代码，是一种软件发布模式，在共享软件的同时开放其源代码以供阅读和修改。Python 亦是如此，可以阅读和改

进源代码，也可以自由发布其相应软件的副本。Python 因为其开源的本质，能够移植到许多平台上，包括了 Windows、Linux、MacOS 等。

(4) 库资源丰富。

Python 不仅继承了 C 语言的优点，能够调用 C 库，同时因为修改自 shell，Python 也有“胶水”语言的优点，有丰富的 API 和标准库。能联结其他语言完成各种功能，包括正则表达式、数据库、网页、电子邮件、XML 等。同时，众多的开源科学计算软件包都提供了 Python 接口，如 OpenCV、NumPy、matplotlib、TensorFlow 等，有了这些库的支持，很多能在 MATLAB 中实现的功能都可以用 Python 来轻松代替，且更轻量、更易维护。

(5) 解释型语言。

将高级语言翻译为机器语言通常有两种方式：编译执行和解释执行。编译执行指源代码由编译器编译为可执行的机器码后再执行，通常执行效率高；解释执行指源代码被解释器直接读取执行，可以灵活地跨平台。Python 语言写的程序不用编译成二进制代码，而是通过解释器将源代码转换成“字节码”这种中间形式，由虚拟机负责运行，即从源代码直接运行程序，所以 Python 是一类解释型语言，便于移植到不同平台。除此之外，Python 还能以交互模式运行，即直接在命令模式中下达操作指令即可运行。

1.1.3 Python 与其他语言

Python 是一种面向对象的编程语言，在 C 语言的基础上有了扩展和提高。下面介绍 Python 与目前比较流行的几种编程语言的关系与区别。

1.1.3.1 Python 与 C 语言

Python 是在 C 语言的基础上发展而来的，但是与 C 语言又有诸多区别：

(1) C 语言采用编译执行方式，执行效率高，不能跨平台，通常用作底层开发；Python 采用解释执行方式，执行速度慢，但易于跨平台。

(2) C 语言是静态编译语言，变量使用前必须先定义，类型不能变化；Python 是动态编译语言，变量的使用只需要赋值即可。

(3) C 语言中数据类型与 Python 中数据类型稍有区别，如 Python 中有 String（字符串）、List（列表）、Sets（集合）、Dictionary（词典）等类型，而 C 语言中是没有这些数据类型的。

(4) C 语言是面向过程的语言，许多功能需要手动编程实现；Python 是面向对象的语言，数据的封装性很好且代码的可重用性很高。

(5) C 语言中使用 {} 来表示语句体；Python 中使用缩进表示语句体。细节上还有 C 语言中 for 的使用与 Python 中 for 的使用的不同。

1.1.3.2 Python 与 C++

Python 与 C++ 同为面向对象的编程语言，也都是在 C 语言的基础上发展而来的，而且 C++ 和 Python 都支持多范式编程，但它们之间仍然存在区别：

(1) C++ 采用编译执行的方式，执行效率较高，常用于开发对效率要求高的软件；

而 Python 采用解释执行，常用于开发小工具、网站和小游戏。

(2) C++ 中也使用 `{}` 来表示语句体，且需要声明定义数据类型（与 C 语言类似）；Python 使用缩进来表示语句体，并且 Python 的语法比 C++ 更简单、更易于使用。

(3) 在垃圾回收方面，C++ 需要程序员手动回收，而 Python 有自己的垃圾回收机制。

1.1.3.3 Python 与 Java

Python 与 Java 都支持面向对象编程，Java 采用严格面向对象编程方法，同时有很多大型的开发框架，比较适合企业级应用。Java 和 Python 都具有自己的垃圾回收机制，且它们都适合程序员协同开发。但 Python 和 Java 还是有些微的不同：

(1) Java 是静态编译语言，变量使用前必须定义；Python 是动态编译语言，使用变量赋值即可。并且 Java 需要编译之后才能运行，但是 Python 可以直接运行。

(2) Java 运行效率要高于 Python，且 Java 的稳定性更高。

(3) Java 中使用 `{}` 来表示语句体，Python 使用缩进来表示语句体。Java 中要使用分号来表示语句结束，Python 中可以不使用。

1.2 Python 的开发环境

“工欲善其事，必先利其器”。本节介绍如何配置 Python 的开发环境，并介绍一些常用的编辑器以供选择使用。

1.2.1 在 Windows 系统下安装 Python

在 Windows 系统下安装 Python 的步骤如下：

Step 1：从官网下载最新的 Python 的安装包进行安装。

在浏览器中输入 <https://www.python.org/> 进入官网，从下载页中下载最新的 Python 版本。图 1-3 为下载页，目前最新的版本为 Python 3.7.2。进入相应下载页之后，选择电脑适合的 executable installer 版本进行下载，如图 1-4 所示。



图 1-3 Python 官网的下载页

Version	Operating System	Description	MD5 Sum	File Size	60%
Clipped source subset	Source release		0247503A70B74C7738132D05A440	2287192	96
X2 compressed source subset	Source release		450C96B189C05B564C293E725476	1749423	145
macOS 64-bit installer	Mac OS X	for Mac OS X 10.6 and later	408D79735400000564C293E725476	3445674	556
macOS 64-bit updater	Mac OS X	for OS X 10.9 and later	F0534970A0481709105B13084840	1776099	145
Windows Setup file	Windows		944579762484000564C293E725476	4801247	755
Windows x86-x64 installation log file	Windows	for AMD64/IA64/IA32	813000000564C293E725476	702385	96
Windows x86-x64 installation metadata	Windows	for AMD64/IA64/IA32	72581670000564C293E725476	3643476	551
Windows x86-x64 install metadata	Windows	for AMD64/IA64/IA32	46612230000564C293E725476	1347688	391
Windows x86-x64 installation log file	Windows		302540000564C293E725476	451926	96
Windows x86-x64 installation metadata	Windows		302540000564C293E725476	2212614	391
Windows x86-x64 install metadata	Windows		302540000564C293E725476	1216448	391

图 1-4 Python 3.7.2 版本下载选择

Step 2: 添加 Python 到环境变量中。

点击 exe 文件进行安装，在安装界面选择添加路径到 Path，在安装时会自动将 Python 路径添加到环境变量中去。安装选择如图 1-5 所示。



图 1-5 安装 Python 3.7.2

当然也可以手动添加 Python 路径。按照默认路径安装是可以自动添加到环境变量中的，但是如果不按照默认路径安装，就需要手动配置环境变量了。首先找到安装路径，然后右键“我的电脑”→属性→高级系统设置→环境变量→找到系统变量中的 Path→编辑 Path，将 Python 路径添加进去→确认即可。添加路径如图 1-6 所示，这里的 Python 路径为 C 盘的 Python37 文件夹，注意不同路径之间使用英文分号间隔。

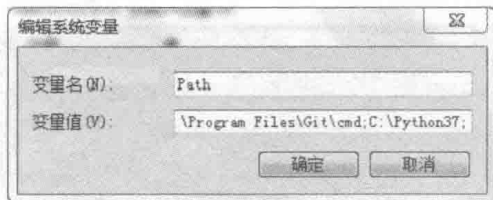


图 1-6 添加 Python 到环境变量

Step 3: 测试 Python 是否安装成功。

键盘上按下 Win+R，在输入框中输入 cmd 打开命令行，再输入 Python 命令，如果出现 Python 的相应版本号，则表示安装成功，如图 1-7 所示。

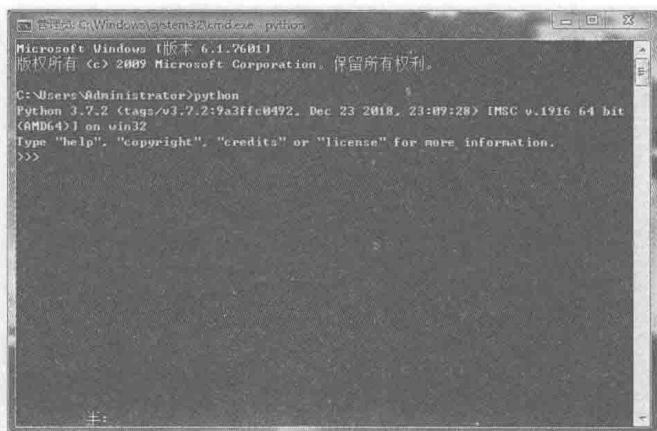


图 1-7 测试 Python 3.7.2 安装成功

1.2.2 在 Mac OS X 下安装 Python

在 Mac 系统下已经有自带的 Python 了，但是自带的版本为 Python 2.X。如果需要启动 Python 2.X，则在终端输入 Python 命令即可启动操作。如果想要使用 Python 3.X 版本，则需要手动安装。

Step 1: 从官网下载需要的 Python 版本的安装文件。

同样的，进入 Python 官网之后再进入相应下载页，选择需要的 Python 版本，然后下载其对应的 Mac OS X 文件。

Step 2: 安装文件，一直点击继续安装完成即可，如果想确认是否安装成功，则使用终端输入 `python3 -V` 命令来查看版本即可。

Step 3: 在终端输入 `python3` 命令即可启动使用 Python 3.X。

1.2.3 Python 编辑器

Python 发展至今，支持 Python 编辑的编辑器也是数不胜数，Python 除了可以直接在命令行执行之外，还自带有 IDLE 编辑器，另外还有许多功能强大的第三方编辑器，如 PyCharm、Notepad++ 等。

(1) 命令行和 IDLE。

Python 可以直接使用命令行进行编辑，但需要注意的是，使用命令行只能逐条输入语句，且不能重复执行。所以使用命令行编辑只适合测试少量简单的代码，不适合复杂程序的编写设计。

使用命令行完成第一行代码的编写：

```
print("Hello World!")
```

图 1-8 为最终测试结果。