中国电子教育学会高教分会推荐
高等学校新工科应用型人才培养“十三五”规划教材
浙江省普通高等学校新形态教材

Python Programming with Big Data

大数据的Python基础

林勇 编著

西安电子科技大学出版社
http://www.xduph.com

中国电子教育学会高教分会推荐

高等学校新工科应用型人才培养“十三五”规划教材

浙江省普通高等学校新形态教材

大数据的 Python 基础

Python Programming with Big Data

林 勇 编著

西安电子科技大学出版社

内 容 简 介

Python 作为一种解释型的编程语言，具有简单、优雅的设计风格和功能强大的扩展库，非常适合海量数据的处理，已成为各行业大数据技术革新的基础性工具。本书系统讲解 Python 的基础知识和程序设计方法，全书共 15 章。第 1～8 章为基础篇，介绍程序设计的基本控制结构，以列表、元组、字典、集合为代表的各类数据结构，函数、字符串与正则表达式的使用方法，面向对象方法、文件及异常处理等基础知识；第 9～15 章为进阶篇，介绍类的成员访问方法、迭代器与生成器，采用 wxPython 的 GUI 编程方法及 NumPy、SciPy、Matplotlib、Pandas 的科学计算与可视化方法，同时采用进程、线程、协程实现并发处理，利用 DB-API、ORM 对象关系映射实现数据库编程。此外，网络程序设计部分主要介绍网络爬虫、Web 应用开发，大数据处理部分介绍函数式编程与 MapReduce 模型。

本书涵盖 Python 程序设计的基本原理、基础知识并提供丰富的程序设计案例，信息量大、知识点紧凑、实用性强。全书的编写围绕大数据的时代特征，紧扣 Python 编程理论和技术发展趋势，可作为高等院校各专业的 Python 语言教材，也可作为 Python 应用开发人员及编程爱好者的参考资料。

图书在版编目(CIP)数据

大数据的 Python 基础 / 林勇编著. —西安：西安电子科技大学出版社，2020.4
ISBN 978-7-5606-5625-0

Ⅰ. ① 大… Ⅱ. ① 林… Ⅲ. ① 软件工具—程序设计—高等学校—教材 Ⅳ. ① TP311.561

中国版本图书馆 CIP 数据核字(2020)第 042269 号

策划编辑 李惠萍
责任编辑 何雪梅 雷鸿俊
出版发行 西安电子科技大学出版社(西安市太白南路 2 号)
电 话 (029)88242885 88201467 邮 编 710071
网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com
经 销 新华书店
印刷单位 陕西天意印务有限责任公司
版 次 2020 年 4 月第 1 版 2020 年 4 月第 1 次印刷
开 本 787 毫米×1092 毫米 1/16 印 张 20.25
字 数 478 千字
印 数 1～3000 册
定 价 45.00 元

ISBN 978-7-5606-5625-0/TP

XDUP 5927001-1

《大数据的 Python 基础》

编 委 会 名 单

编　著　林　勇

参　编　杨　帆　　汪　保　　滕　宇　　盛晓春

尹　志　　陆星家　　尹天鹤　　杨　芳

陈志荣　　王书琦　　宋加涛　　胡大威

前　言

Python 是一门拥有简洁语法和高效信息处理能力的解释型高级编程语言，采取免费、开源的方式进行维护和管理，除拥有大量功能强大的内置对象、标准库、扩展库之外，还允许各行各业的科技人员结合本领域的需要进行专门的设计和扩充，从而极大地提高了 Python 语言本身的活力和适用范围。对于大数据系统这样涵盖内容丰富、数据容量巨大又需要广泛算法支撑的业务和应用系统，Python 语言的使用具有得天独厚的优势，可以方便地进行编程并使用其各类相关的功能模块，提高数据分析处理的能力。

Python 语言的设置充分融入了现代高级编程语言的精髓，可以通过简单的语句实现以往需要编写大量代码才能够实现的功能。对于常用的数据结构、函数、字符串和正则表达式、面向对象方法、文件操作以及异常处理等基础性程序设计方法和用法均支持。本书侧重于从基础性原理和方法出发，循序渐进、由浅入深地引导学习者进行 Python 语言知识与编程方法的学习，并在学习基本编程方法的基础之上，进一步学习类的成员访问方法、迭代器和生成器等知识，进而学习 GUI 编程方法、科学计算与可视化方法、并发编程的相关知识与原理、数据库编程、网络程序设计以及大数据处理方法。

· 本书的内容组织

本书共 15 章，分基础篇和进阶篇两个部分。第 1～8 章为基础篇，主要内容包括 Python 语言概述，程序设计基础，序列结构，函数，字符串与正则表达式，面向对象程序设计，文件操作，异常处理与程序调试；第 9～15 章为进阶篇，主要内容包括成员访问、迭代器与生成器，GUI 编程，科学计算与可视化，并发编程，数据库编程，网络程序设计，大数据处理。

本书主要有以下几个特点：

(1) 涵盖 Python 程序设计的基本原理、基础知识并提供丰富的程序设计案例，信息量大、知识点紧凑、实用性强。

(2) 围绕大数据的时代特征，紧扣 Python 编程理论和技术发展趋势，对知识脉络的设计符合国际国内对高质量 Python 程序设计人员的需要。

(3) 注重对核心技术及基础性原理与方法的讲解，让学习者能够举一反三，深入了解现代程序设计的精髓与先进理念，更有利于对实际问题的领会和解决。

(4) 本书给出了丰富的案例和练习实例，有利于学习者对知识的理解和深入领会，能够辅助学习者掌握关键知识点。

· 本书的读者对象

本书适用于相关专业的本科生或研究生，供各类大专院校和职业技术学院作为教材或教学参考书使用，也可作为相关培训课程的教学资料；本书也可供相关软件开发及科技人

员作为辅助学习资料与工作参考材料使用，对有兴趣学习 Python 编程技术的人员或各类相关程序设计竞赛参赛人员也具有一定的参考价值。

采用本书作为教材时，学时安排参考如下：

(1) 本书作为计算机、数据工程、信息技术、电子、自动化、人工智能、大数据等相关专业本科或研究生的程序设计教材时，建议采用 64 或 72 学时，讲授本书的全部章节，也可结合专业特点及学时安排，选取第 1～9 章作为必讲章节，第 10～15 章作为选讲章节。

(2) 本书作为会计、经济、金融、管理、统计以及其他非工科专业的研究生或本科生教材时，建议采用 64 学时，选取第 1～9 章作为必讲章节，第 10～15 章作为选讲章节。

(3) 作为非计算机相关专业的本科生公共基础课程序设计教材时，建议采用 48 或 64 学时，选取第 1～8 章作为必讲章节，第 9～15 章作为选讲章节。

(4) 作为专科院校或职业技术学院程序设计教材时，建议采用 64 或 96 学时，可结合专业特点及学时安排讲授本书的全部章节，或选取第 1～8 章作为必讲章节，第 9～15 章作为选讲章节。

(5) 作为 Python 培训用书时，建议培训时间为 7～12 天，可结合培训学时安排讲授本书的全部章节，或选取第 1～8 章作为必讲章节，第 9～15 章作为选讲章节。

·本书的配套资源

本书配备多媒体教学资料，相关例题和一些必要资料可以直接通过书中二维码扫码查询。为方便教学，本书提供全套教学课件、全书例题的源代码、参考教学大纲、学时分配表以及试题样卷等资料，可向西安电子科技大学出版社索取，或在其官网(http://www.xduph.com)自行查询。本书也开放了课后习题的参考答案，有需要的老师请直接联系西安电子科技大学出版社获取。

本书被认定为高等学校新工科应用型人才培养“十三五”规划教材和浙江省普通高等学校新形态教材，出版过程中得到了宁波工程学院、浙江大学等院校师生和西安电子科技大学出版社、浙江大学出版社的鼎力支持和帮助，在此表示衷心的感谢。由于编者水平有限，书中难免有错漏之处，恳请广大读者不吝指出并提出宝贵意见，我们将在今后再版时修订完善。

编　者

2020 年 1 月 10 日

目　录

第一部分　基　础　篇

第二部分 进 阶 篇

第一部分 基础篇

- Python 语言概述
- 程序设计基础
- 序列结构
- 函数
- 字符串与正则表达式
- 面向对象程序设计
- 文件操作
- 异常处理与程序调试

第 1 章　Python 语言概述

21 世纪是数据信息时代，移动互联、社交网络、电子商务大大拓展了互联网的疆界和应用领域，随之而来的是各种数据及数据量的急剧膨胀，于是人们引入大数据(Big Data)一词来描述和定义信息爆炸所产生的海量数据。随着人工智能、大数据技术的发展，Python 逐步成为 C、C++、Java 之外最受欢迎的编程语言，其简单、易用和功能强大等特点，使其成为适用于数据科学和大数据技术的专业化编程工具。本章将对大数据的特点和 Python 语言的发展以及基本使用进行广泛深入的探讨。

1.1　大数据的时代特征

随着时间的推移，人们越来越多地意识到数据的重要性。大数据时代对人类的数据驾驭能力提出了新的挑战，也为人们获得更为深刻、全面的洞察能力提供了前所未有的空间与潜力。哈佛大学社会学教授加里·金说："这是一场革命，庞大的数据资源使得各个领域开始了量化的过程，无论学术界、商界还是政府，所有领域都将展开这种过程。"进一步来说，大数据是指那些超过传统数据库系统处理能力的数据。它的数据规模很大，对传输速度要求很高，其结构已大不同于原本的数据库系统。为了获取大数据中的价值，我们必须选择另一种方式来处理它。数据中隐藏着有价值的模式和信息，在以往，需要大量的计算时间和运算成本才能提取这些信息，而当今不断更新的计算机软硬件以及云计算等资源使得大数据的处理更为方便和廉价，这样，企业和普通民众都能够享有大数据时代所带来的诸多便利。

以电子商务为例，零售业中通过对门店销售、地理和社会信息的分析能提升对客户信息和客户数量的把握程度，进而根据用户的情况实行有原则和有目的的合理化商业布局。在社交网络领域，Facebook 通过结合大量用户信息，定制出高度个性化的用户体验，并创造出新型的广告模式。这种通过大数据创造出新产品和服务的商业行为并非巧合，谷歌、雅虎、亚马逊、Facebook 以及国内的阿里巴巴、百度、淘宝、京东、QQ、微信等平台都是大数据时代的创新者。

大数据的特征可以归结为以下四点：

(1) 海量性(Volume)：企业面临着数据量的大规模增长。例如，IDC 最近的报告预测称，到 2020 年，全球数据量将扩大 50 倍。目前，大数据的规模尚是一个不断变化的指标，单一数据集的规模范围从几十太字节(TB)到数拍字节(PB)不等。而随着互联网以及移动电子商务、人工智能等技术的不断实用化，各种意想不到的来源都能产生数据。

(2) 多样性(Variety)：由于各类系统会产生海量业务数据，而网络日志、社交媒体、互

联网搜索、手机通话记录及传感器网络等各类数据源也会产生品种繁多的数据。此外，大量图片、声音和视频文件的传播，也是造成网络数据多样性的因素之一。

(3) 高速性(Velocity)：高速描述的是数据被创建以及被传播的速度。随着芯片等技术的不断优化，电脑和手机的处理能力不断提高，展望即将到来的量子计算和第五代移动通信网络(5G)等技术，可以预期今后的网络和设备将以更高的频率创建新型数据，同时以更快的速度传播这些数据，满足人类日益提高的数据需求。

(4) 价值易变性(Value)：大数据具有多层结构，这意味着大数据会呈现出多变的形式和类型。相较传统的业务数据，大数据存在数据类型不规则、数据定义不统一等特点，造成很难甚至无法使用传统的应用软件进行分析。因此，企业面临的挑战是处理并从各种形式呈现的复杂数据中挖掘出有价值的信息。

大数据的这四个特点因其英文首字母都为 V 又简称为大数据的 4V 特征。这些特征给计算机软件和编程技术的发展带来了新的挑战，必须不断寻求更加高效、便捷的编程处理手段，才能够有效地适应大数据时代的技术挑战。在这一背景下，Python 语言作为新生代的编程工具得到了前所未有的爆发。简洁、开源是这款工具吸引众多程序设计人员的原因，这使得 Python 编程成为数据分析和挖掘的一种便利工具，不仅在教育、科研等领域中迅速得以推广，也在广大行业应用领域得以实际应用，解决众多国家和社会的实际问题。

1.2 Python 语言的发展

Python 是一门优雅而健壮的编程语言，它继承了传统编译语言的强大性和通用性，同时也借鉴了简单脚本语言和解释语言的易用性。简单性为大量代码的编写和阅读提供了便利，而代码的简单又并不失通用性与强大性等特征，使得原来利用 C、C++、Java 等编程语言需要编写几十甚至上百行代码的程序，在 Python 中可能仅仅利用几行程序就能实现相同的功能。这些特点为大数据的便捷分析和处理提供了可能。

1.2.1 版本更迭

荷兰的吉多·范罗苏姆(Guido van Rossum)于 1989 年底为了能够访问分布式操作系统的系统调用，创建了一种通用的程序设计语言，这就是 Python 的雏形。现如今，应用开发工程师、运维工程师、数据科学家都喜欢 Python，使得 Python 成为大数据系统的全栈式开发语言。Python 不像 C 或 C++一样需要做很多的底层工作，它可以快速进行模型验证，且语法简洁、表达能力强，可以通过较少的代码实现 C 语言、Java 语言等需要大量程序代码才能够实现的功能。与 Matlab、R 语言等科学计算领域的编程语言相比，Python 不但能够实现等价的科学计算能力，还能够方便地植入到行业应用之中直接用于解决实际问题。

当前的 Python 已经进入了第三次版本的更迭，即 Python 3。Python 3 于 2008 年年末发布，解决和修正了以前语言版本的内在设计缺陷。Python 3 开发的重点是清理代码库并删除冗余，清晰地表明只能用一种方式来执行给定的任务。

在 Python 3 发布之后又出现了 Python 2.7 版本，其目的在于通过提供兼容性的措施，使 Python 2.x 的用户更容易将功能移植到 Python 3 上。虽然 Python 2.7 和 Python 3 有许多类似的功能，但它们不应该被认为是完全可互换的。在代码语法和处理方面 Python 2 和

Python 3 将会有较大的差异，主要包括以下两个方面。

1. print 语句

在 Python 2 中，print 被视为一个语句而不是一个函数，这是一个典型的容易混淆的地方，因为在 Python 中的许多操作都需要括号内的参数来执行。如果在 Python 2 中要在控制台输出“hello world”，则输入以下 print 语句：

```
print " hello world "
```

在使用 Python 3 时，print()会被显式地视为一个函数，因此要输出上面相同的字符串，需要使用以下的函数调用语法：

```
print("hello world ")
```

这种改变使得 Python 语法更加一致，并且在不同 print 函数之间进行切换更加容易。

2. 整数的除法

当进行数值计算时，我们往往希望计算机能够得到像数学方式计算出的答案，比如：

```
5 / 2 = 2.5
```

然而在 Python 2 中，整数是强类型的，不会被看成是带小数点的类型(称为浮点数)。因此 Python 2 的程序并不能够得到以上结果。当除法符号/的任一侧的两个数字是整数时，Python 2 进行底除法，使得对于商 x，返回的数字是小于或等于 x 的最大整数，在这种情形下，我们输入以下程序语句：

```
a = 5 / 2
print a
```

根据以上原则，其输出结果为：

```
2
```

为解决这个问题，可以添加小数位，以得到预期的答案 2.5，比如：

```
a = 5.0 / 2.0
print a
```

此时的输出结果为：

```
2.5
```

在 Python 3 中，整数除法变得更直观，如

```
a = 5 / 2
print(a)
```

此时的输出为：

```
2.5
```

在 Python 3 中的这种调整使得整数除法能够更为直观地符合使用者的预期。从长远的情况看，Python 3 代表了今后 Python 的发展方向，因此在本书后续章节中，如无特殊说明，将以 Python 3 编程规范为主进行介绍。一般情况下，在 Python 2.7 中可以直接运行或者进行少量调整后直接支持利用 Python 3 规范所编写的程序。

1.2.2 软件实现

Python 作为一门解释型的编程语言，程序源码不需要编译，而是由 Python 解释器将源

代码转换为字节码，再由 Python 解释器来执行这些字节码。

符合 Python 语言规范的解释程序以及标准库有多种软件实现，不同实现方法之间有一定差别。下面分别列出几个主要的实现。

(1) CPython：是 Python 的官方版本，使用 C 语言实现，使用最为广泛，新的语言特性一般也最先出现在这里。CPython 实现会将源文件(py 文件)转换成字节码文件(pyc 文件)，然后运行在 Python 虚拟机上。

(2) Jython：是 Python 的 Java 实现，相比于 CPython，它与 Java 语言之间的互操作性要远远高于 CPython 和 C 语言之间的互操作性。在 Python 中可以直接使用 Java 代码库，这使得使用 Python 可以方便地为 Java 程序写测试代码。Jython 会将 Python 代码动态编译成 Java 字节码，然后在 JVM 上运行转换后的程序，这意味着此时 Python 程序与 Java 程序没有区别，只是源代码不一样。

(3) Python for.NET：实质上是 CPython 实现的.NET 托管版本，它与.NET 库和程序代码有很好的互操作性。

(4) IronPython：是一种在.NET 及 Mono 上的 Python 实现，基于微软的 DLR 引擎。IronPython 并未实现 Python 通用类库，仅实现了部分核心类。

(5) PyPy：是 Python 的 Python 实现版本。PyPy 运行在 CPython(或者其他实现)之上，用户程序运行在 PyPy 之上。它的一个目标是成为 Python 语言自身的试验场，因为可以很容易地修改 PyPy 解释器的实现(因为它是使用 Python 编写的)。

(6) Stackless：CPython 的一个局限就是每个 Python 函数调用都会产生一个 C 函数调用。这意味着同时产生的函数调用是有限制的，因此 Python 难以实现用户级的线程库和复杂递归应用。一旦超越这个限制，程序就会崩溃。Stackless 的 Python 实现突破了这个限制，一个 C 栈帧可以拥有任意数量的 Python 栈帧。这样就能够拥有几乎无穷的函数调用，并能支持巨大数量的线程。Stackless 的问题是它要对现有的 CPython 解释器做重大修改，因而它属于一个独立的分支。

1.3　Python 开发环境配置

进行 Python 程序设计之前，首先应搭建其开发环境，并建立起程序的运行平台。有两种 Python 的安装方式，一种是采用 Python 官方文件安装 Python 软件，另一种是利用 Anaconda 包管理器安装和管理 Python 软件，以下分别进行介绍。

1.3.1　Python 的安装和运行

直接安装 Python 软件具有很大的灵活性，可以根据需要自行选择和配置 Python 安装与运行环境，需要使用扩展模块时可以自行运用 pip 包管理工具随时安装或卸载，属于标准的 Python 包管理方式，适用于初学者和高级开发人员。

进入 Python 官网 https://www.python.org/，下载合适的 Python 安装包，以下选择的是 Python 3.6.5，一般直接选择最新版本即可。完成安装包的下载后，双击下载的 exe 文件进行安装。

运行安装包后进入图 1-1 所示的安装界面，可以直接选择 Install Now 进行安装，也可以选择定制化安装 Customize installation。一般情况下，对于开发和学习用户来说，在可选特征中可以选择全部特征，方便今后的使用。安装界面中的 Add Python 3.6 to PATH 选项是将 Python 的软件目录添加到系统的可执行文件目录变量 Path 中，方便直接在命令行启动 Python，在此可以选择，也可以安装完成后手工将 Python 的安装路径直接添加到系统 Path 变量的路径之中。安装完成后，可以在操作系统的启动菜单中找到 Python 的快捷方式，如图 1-2 所示。

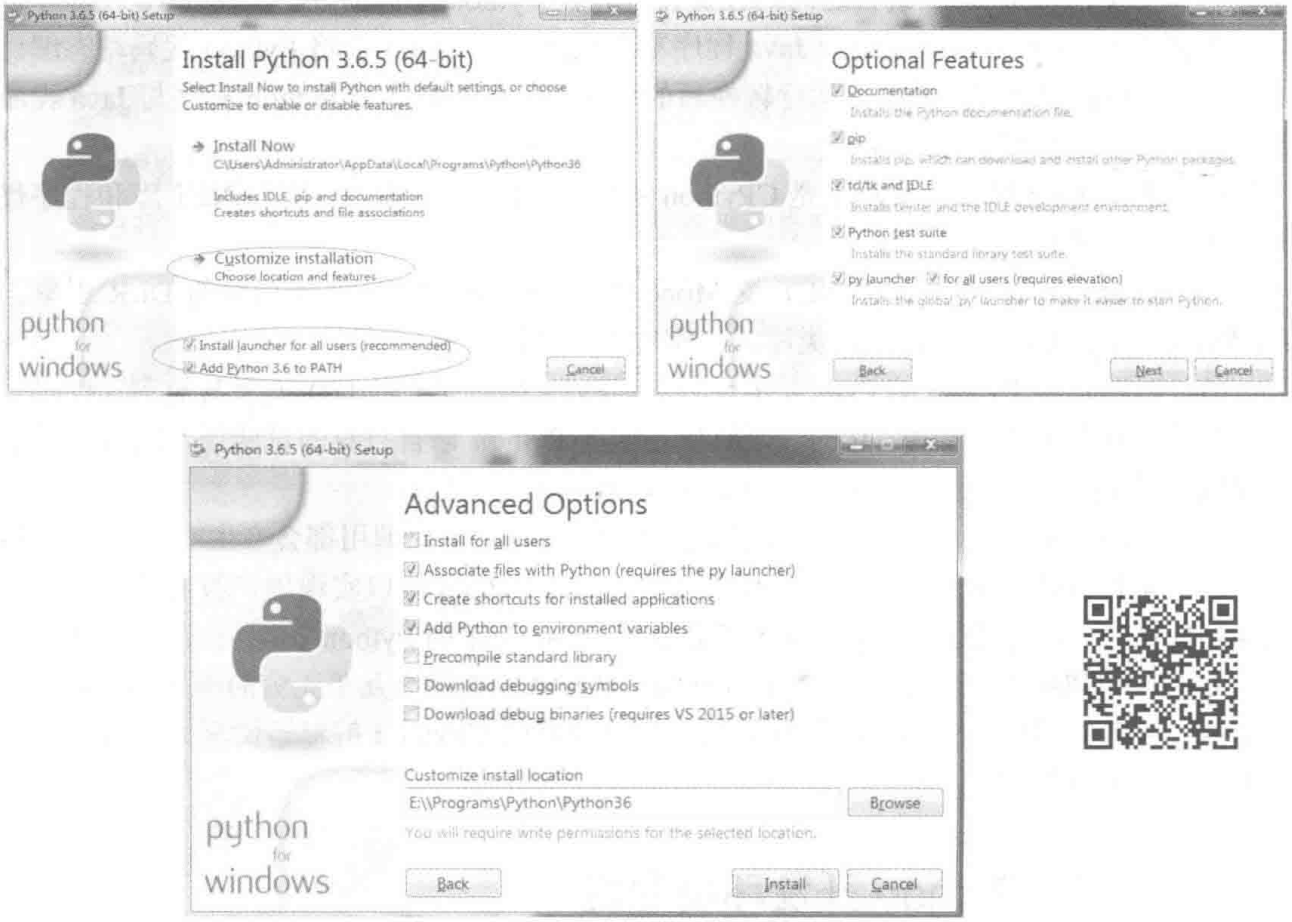

图 1-1　Python 软件安装

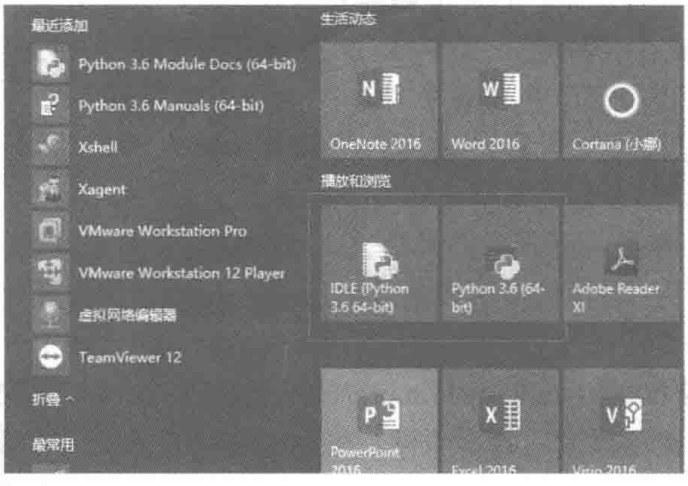

图 1-2　Python 的快捷启动方式及其终端窗口

点击 IDLE (Python 3.6 64 bit)或者 Python 3.6(64 bit)，即可通过 Python 自带的集成开发工具 IDLE 和终端窗口开始使用 Python。如果安装过程中选择了将 Python 的安装路径添加到系统环境变量的可执行路径 Path 中，也可以在安装完成后使用 Win+R 组合键，输入 cmd 打开命令提示符窗口，再输入 python 进入交互式终端。在系统的 Path 环境变量中添加 Python 一般只需要添加两个路径，如 Python 的安装路径为 C:\Program Files\Python36，则只需在 Path 环境变量中添加 C:\Program Files\Python36 和 C:\Program Files\Python36\Scripts 两个路径。

进入 Python 终端后即可看到“>>>”的交互式提示符，此时即可以程序语句的形式与系统进行交互。如图 1-3 所示，输入 print("hello world")，能够看到系统显示“hello world”。

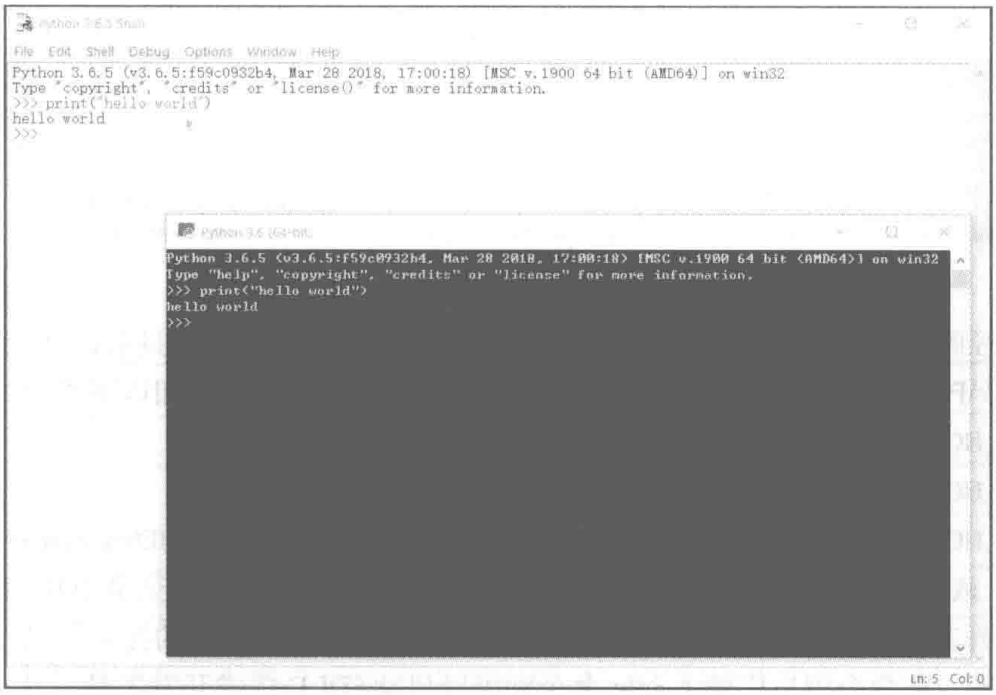

图 1-3　Python 的快捷启动方式及其终端窗口

使用 Python 内置的交互式解释器，可以配合文本编辑软件(如 Windows 记事本 Notepad.exe)进行程序的编写。也可以直接使用一些集成开发环境，如 Eclipse+Pydev 插件，这样就能够直接实现 Python 的集成化开发与运行环境。

本书的讲解以这种标准的 Python 安装、配置和运行方式为基础，利用 pip 包管理器安装或卸载 Python 的扩展模块，利用 IDLE 结合 DOS 命令行窗口进行程序运行或调试。

1.3.2　Anaconda 包管理器的使用

另一种 Python 软件的安装方式是采用 Anaconda 集成包管理器。Anaconda 的优点是已经内置了 Jupyter notebook 交互式计算环境、Spyder 集成开发环境，以及 NumPy、Pandas、Matplotlib、SciPy 等各类主要数据分析模块。安装 Anaconda 后，很多工具和模块可以直接使用，具有一定方便性。

要安装 Anaconda 软件，可以在其网站(https://www.anaconda.com/distribution)下载最新版本的安装包，如 Python 3.7 版本的 Anaconda3(注意 Anaconda2 支持的是 Python 2 编程规范)。选择对应的操作系统版本即可安装，期间需要选择安装路径并进行系统设置(见图 1-4)。