HZ BOOKS
华章科技

The Pragmatic Programmers

华章专业开发者丛书

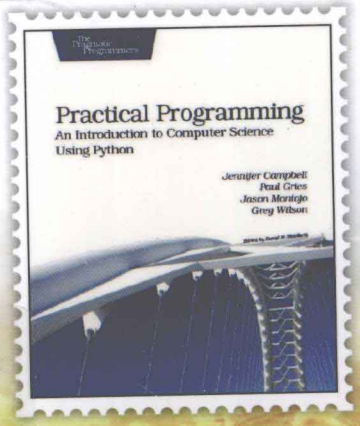
Python编程入门读物
《代码之美》编者最新力作

Python 编程实践

Practical Programming
An Introduction to Computer Science Using Python

（美）Jennifer Campbell
Paul Gries
Jason Montojo
Greg Wilson
著

唐学韬 等译



机械工业出版社
China Machine Press

华章专业开发者丛书

Python编程入门读物
《代码之美》编者最新力作

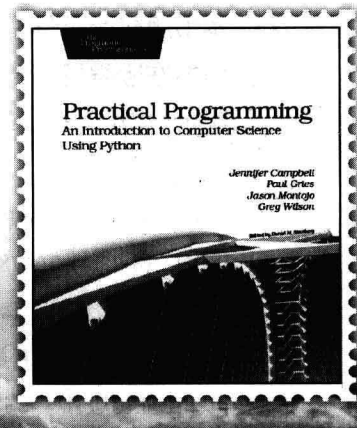
Python

编程实践

Practical Programming
An Introduction to Computer Science Using Python

(美) Jennifer Campbell
Paul Gries
Jason Montojo
Greg Wilson 著

唐学韬 等译



机械工业出版社
China Machine Press

Python是当今世界流行的编程语言之一。本书共15章，通过一些短小精悍的交互式Python脚本帮助学生进行练习，并在这个过程中掌握诸如数据结构、排序和搜索算法、面向对象编程、数据库访问、图形用户界面等基本概念以及良好的程序设计风格。本书既是一本注重科学的计算机科学专业教材，也是一本目标明确的Python参考书。

本书语言风格言简意赅，图表丰富，简单实用，是一本优秀的Python入门级读物，适合Python初学者使用。

Jennifer Campbell, Paul Gries, Jason Montojo, Greg Wilson. Practical Programming : An Introduction to Computer Science Using Python（ISBN 978-1-934356-27-2）.

本书版权登记号：图字：01-2011-6287

图书在版编目（CIP）数据

Python编程实践/（美）坎贝尔（Campbell, J.）等著；唐学韬等译. —北京：机械工业出版社，2012.1

（华章专业开发者丛书）

书名原文：Practical Programming : An Introduction to Computer Science Using Python

ISBN 978-7-111-36478-8

I. P…　II. ①坎…　②唐…　III. 软件工具—程序设计　IV. TP311.56

中国版本图书馆CIP数据核字（2011）第234147号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：秦　健

北京京北印刷有限公司印刷

2012年1月第1版第1次印刷

186mm×240mm • 20印张

标准书号：ISBN 978-7-111-36478-8

定价：49.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

对本书的赞誉

这本书中的知识点组织得很好，它是围绕着一些有用的任务而展开的，而不是简单地罗列出一些抽象的概念。同时，每一章都恰到好处地介绍了 Python 编程的某个重要方面。对于想要知道“我该如何让计算机做 XXX”的学生而言，本书可以让其轻而易举地找到答案。

——Christine Alvarado

哈维玛德学院计算机科学副教授

科学其实就是在实践中不断学习的过程。这本书鼓励计算机科学专业的学生通过一些短小精悍的交互式 Python 脚本进行练习，并在这个过程中掌握诸如数据结构、排序和搜索算法、面向对象编程、数据库访问、图形用户界面等基本概念以及良好的程序设计风格。‘言简意赅的文字，再配以大量有趣的例子、图表，使得这本书成为一本绝佳的入门级读物。

——Ronald Mak

IBM Almaden 研究中心研究员

圣何塞州立大学计算机科学系讲师

什么？没介绍编译器，也没有用两个完整的应用程序做例子？这到底是什么编程类图书啊？告诉你吧，这属于很棒的那种。它首先假设“你什么也不懂”，并让你在开始学习的时候轻松愉快地长驱直入，然后再把相关的编程技术的概念和理论一一介绍给你。学生们在练习的时候对着那些各种各样的图片肯定会觉得很有意思！

——Laura Wingerd

《Practical Perforce》一书作者

这本书关于调试的那部分内容实在是太棒了！我认识很多程序员初学者，如果你建议

他们看完整本书，他们可能会感到不畅快，但他们却会时不时地温习这一部分（还有许多别的内容）。

——Alex Martelli

《Python in a Nutshell》一书作者

这本书在两个方面取得了成功。它既是一本注重科学的计算机科学专业教材，也是一本目标明确的 Python 参考书。它不仅帮学生建立了计算型思维方式，还鼓励学生将自己刚学会的编程技能直接应用到实验室或他们自己的项目中去。

——Zachary Dodds

哈维玛德学院计算机科学副教授

目　录

第1章

引　言

本章内容

- 程序和编程
- 一点说明
- 要安装什么
- 给教师们的话
- 小结

首先看一下图 1-1 中的那两张照片。第一张照片摄于 1975 年，展示的是位于亚马逊流域的森林。第二张所拍的是同一个区域，只不过是 26 年后的景象。随便哪个人都能看出大量的雨林惨遭破坏，但究竟怎样才算是“大量”呢？

接下来再看看图 1-2。

这些血细胞健康吗？从这些血细胞中能不能看出白血病的一些端倪呢？就算是给一位经验丰富的医生，他也得花上好一阵子才能搞清楚这个问题。如果将这点时间乘以需要检查的病人的数量，那么很显然，这世上根本没有足够多的医生能够检查所有病人。

这就是计算机大展拳脚的地方。计算机程序能够测算两张照片之间的区别，也能够数出血液样本中那些形状异常的薄片的数量。遗传学家通过计算机程序来分析基因序列；统计学家用计算机程序来分析疾病的分布和扩散情况；地质学家用计算机程序来预测地震将会带来的影响；经济学家用计算机程序来分析股市的走势；气候学家用计算机程序来研究全球变暖现象。越来越多的科学家都在编写计算机程序来辅助完成他们自己的工作。而这些计算机程序又反过来催生了许多新的科学门类。

当然了，计算机绝对不只是对科学界有用。我们就是用计算机来编写这本书的；说不定你今天也用到了一台计算机，可能是为了跟朋友们聊天，也可能是为了找到讲座的地点，还可能是为了找到一家售卖披萨和中餐的饭馆。每天都会有人找到让计算机完成之前无法完成的某样事情的办法。总之，所有的那些“某样事情”都在悄然地改变着我们的这个世界。

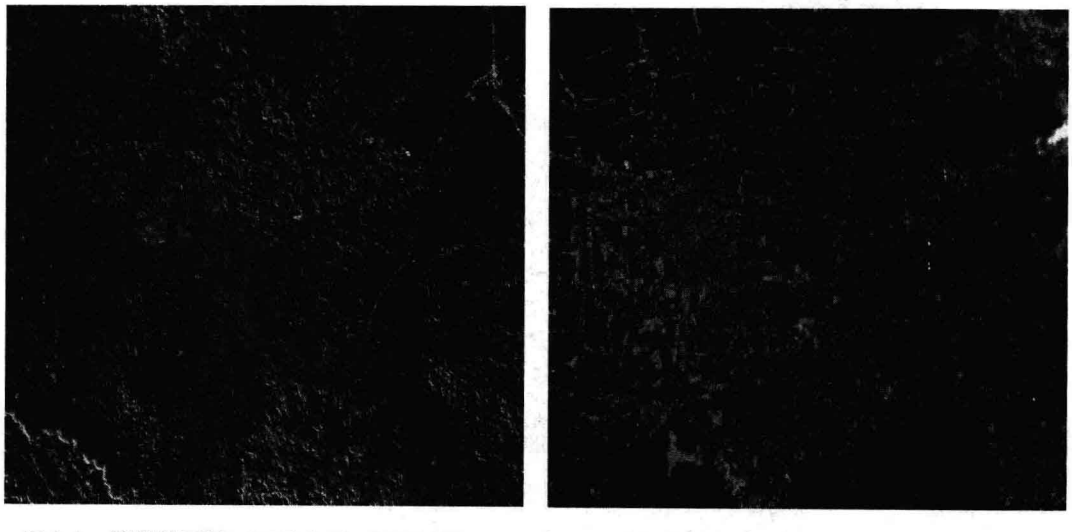


图 1-1　消退的雨林（照片来源：NASA/Goddard Space Flight Center Scientific Visualization Studio）

本书将教会你如何让计算机做你希望它做的事情。也许你只是想成为一名医生、语言学家或是物理学家，而并没有想过要当一名全职的程序员，然而不管你是做什么的，编程的能力都是很重要的，这跟写字和基本算术的能力是一样的。

本章从解释什么是程序和编程开始。然后，我们将会给出一些术语，并为任课教师提供一些虽然无聊但却很有必要的信息。

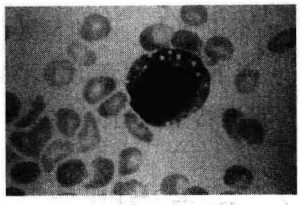


图 1-2 健康的血细胞——是吗？（照片来源：CDC）

1.1 程序和编程

程序其实就是一组指令的集合。当你把去你家的具体路线写给朋友时，你实际上就是在编制一个程序。然后你的朋友就会按照那些指令一条一条地“执行”这个程序。

每个程序都是用一组基本操作指令写出来的，而这些指令都是其读者已经知晓的。比方说，你写给朋友的那组行进指令可能会包含下面这些内容：“在 Darwin 街左转”、“直走三个街区”、“如果你到了加油站的话，赶紧往回走——你走过头了”。

计算机与计算机之间其实没什么区别，只是它们有着不同的操作指令集而已。有些操作是数学方面的，比如“给某个数字加 10，然后再求其平方根”，而有些则可能是关于其他方面的，例如，“从一个名为 data.txt 的文件中读取一行”、“画一个蓝色的像素点”或是“给这本书的作者发个电子邮件”等。

计算机跟老式的计算器之间最大的区别在于：你可以使用老式计算器的相关术语定义出一组新的运算指令，并将这些指令“教给”计算机。比如说，你可以教会计算机“求平均数”就是要“把一组数字全部加起来，然后再除以这组数字的个数”。此外，还可以用你所定义的这些运算指令创造出更多的运算指令，每个新的运算指令都构建于之前所创建的那些指令。这与生命的创造过程类似，许多原子组合在一起构成了蛋白质，然后许多的蛋白质又组成了细胞和长颈鹿。

定义新的操作指令，并将其组合到一起以便能够做一些有意义的事情，这就是编程工作的核心和灵魂。此外，在思考其他类型的问题时，这也是一种非常强大的手段。Jeannette Wing 教授［Win06］认为计算型思维主要包括以下内容：

- 强调概念化，而非程序化。计算机科学不是计算机编程。像计算机科学家一样的思考，不只是说要会编程，还需要能在多个抽象层次上进行思考。
- 它是人类（而非计算机）的一种思考方式。计算型思维是人类解决问题的一种途径，它并不会试图让人类像计算机那样去思考问题。计算机其实是很令人生厌的，人类才是既聪明又富有想象力的。其实是我们人类让计算机变得激动人心的。有了计算机，我们就

可以利用我们的聪明才智去处理那些曾经不敢触碰的问题了，还可以构建功能丰富的计算系统，这一切仅仅受限于我们的想象力而已。

- 不论何人，不论何处。当计算型思维充分融入到人们的工作学习当中，并且如常识般地从我们的注意力中消失时，它就成为了现实。

我们希望，当你读完本书的那个时候，你能从另一种视角来看待这个世界。

1.2 一点说明

专有名词容易让人迷惑的地方是如何称呼某些特定的字符。Python 的语法风格指南（以及许多其他文档）都使用那些符号的原名，所以本书也将这么处理：

() 圆括号（parenthese）

[] 方括号（bracket）

{} 花括号（brace）

1.3 要安装什么

如果想要获取编写本书时的安装指引，请下载本书网站上的那些代码，然后在浏览器中打开 install/index.html。本书网站的网址为 http://pragprog.com/titles/gwpy/practical-programming。

1.4 给教师们的话

本书将通过 Python 这门编程语言来讲述标准的计算机科学主题，此外还将提供很多有用的应用程序。我们之所以选择 Python 主要基于以下几个原因：

- 它是免费的且拥有良好的文档支持。事实上，Python 是目前最大的且组织得最好的开源项目之一。
- 到处都能用。小到手机，大到超级计算机，都在使用其引用实现（由 C 语言编写而成）。此外，在 Windows、Mac OS X 以及 Linux 上都有专业级的安装工具。
- 它有着清晰明了的语法。是的，所有语言都这么说，但是以过去 4 年我们在多伦多大学的实际使用情况来看，我们发现学生们所犯的“标点符号”错误明显比使用类 C 语言要少得多。
- 它变得越来越重要。成千上万的公司每天都会用到它；它是 Google 三大“官方语言”之一；游戏“Civilization IV”中有很大部分代码是使用 Python 写的。此外，它还广泛地被许多高校研究小组所采用。

- 它有着许多优秀的辅助工具。诸如 Vi 和 Emacs 等历史悠久的编辑器都含有 Python 编辑模式，此外还有许多专业级的 IDE。（我们所使用的工具是一款名叫 Wing IDE 的学生免费版。）

我们将使用一种“先对象后类”的教学模式：学生首先将学习如何使用标准库中的对象，而不是从一开始就学习如何创建自己的类，直到他们明白流程控制以及一些基本的数据结构之后再来学习有关类的内容。这将避免许多问题，比如不用再向那些从未接触过编程的学生解释 Java 中的 public static void main(String[] args)。

本书分为两个部分。第一部分介绍了一些基础编程思想：基本数据类型（数字、字符串、列表（list）、集合（set）以及字典（dictionary））、模块（module）、控制流（control flow）、函数（function）、测试、调试以及算法等。根据读者的实际情况，这部分内容可能需要 9 到 10 周时间。

本书的第二部分由一些知识结构上相对独立的章节组成，这些章节主要讲解的是一些较为高级的主题（假设读者已经掌握了全部的基础知识）。这一部分首先告诉学生如何创建自己的类，并讲解何谓封装、继承以及多态（计算机科学专业的课程大都包含这部分知识）。然后讲解一些应用程序方面的知识（比如 3D 图形、数据库、GUI 制作）以及一些基本的 Web 编程知识。编写这部分内容时不仅考虑了计算机科学专业的学生，也考虑了其他科学专业的学生，力求使得本书对他们都有用。

市面上还有许多非常棒的有关 Python 编程的书籍。针对初学者的有 [Guz04，Zel03]，针对具备一定编程经验的有 [DEM02，GL07，LA03]。也许你还想看看 Python 教育特别兴趣小组 [Pyt]，这是一个专门针对 Python 教育工作者的特别兴趣小组。

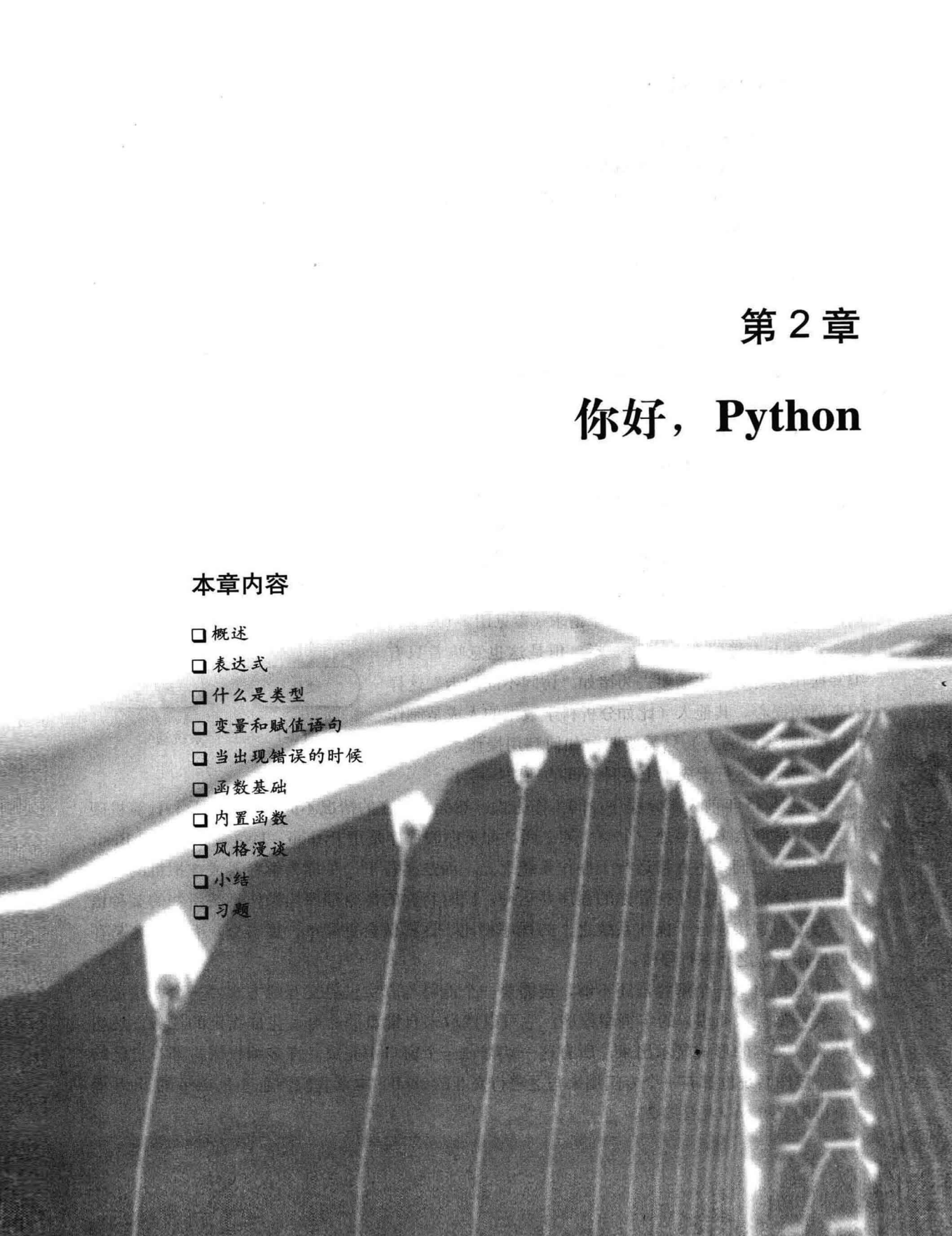
1.5 小结

我们将在本书中做以下几样事情：

- 我们将告诉你如何开发并使用计算机程序来解决真实世界中的问题。书中大部分例子都来自于科学和工程领域，不过编程的思想完全可以应用于其他任何领域。
- 首先，我们将告诉你这门叫做 Python 的编程语言的核心功能。所有的现代编程语言均含有这些功能，因此，不管你今后使用何种编程语言，本书中的知识都能派上用场。
- 我们还将告诉你如何理解编程工作当中的方法论。比如说，我们将阐述如何将复杂问题拆解为一个个简单的问题，然后再将这些简单问题的解决方案组装到一起并以此创建出复杂的应用程序。
- 最后，我们还将介绍一些能够提高编程效率的工具，以及一些能够让你的应用程序解决大型问题的工具。

第 2 章

你好，Python



本章内容

- 概述
- 表达式
- 什么是类型
- 变量和赋值语句
- 当出现错误的时候
- 函数基础
- 内置函数
- 风格漫谈
- 小结
- 习题

计算机程序其实就是由一条条计算机能够看明白的命令组成的。这些命令称为语句（statement），计算机执行的就是这些东西。本章将介绍一些最为简单的 Python 语句，并告诉你该如何用它们来完成一些基本的运算。这些内容本身其实并不是很好玩，不过它们却是我们后面学习的基础。

2.1 概述

为了明白编程究竟是怎么一回事，你首先需要了解“程序是如何在计算机上执行”的一些基本知识。计算机本身是由一块块硬件组装而成的，包括一个用于执行指令和完成运算的处理器、一个用于存储数据的东西（比如硬盘）以及一些其他的东西（比如显示器、键盘、用于连接网络的网卡……）。

为了能够使用这些硬件，所有的计算机都需要运行某种操作系统，比如 Microsoft Windows、Linux 和 Mac OS X 等。操作系统（简称 OS）也是一个程序，但是它很特别，因为它是计算机上唯一能够直接访问硬件的程序。当计算机上的其他程序想要在屏幕上绘制点东西，或是找出按下了键盘上哪个键，或是从硬盘读取数据，它都需要向操作系统发送一个请求（参见图 2-1）。

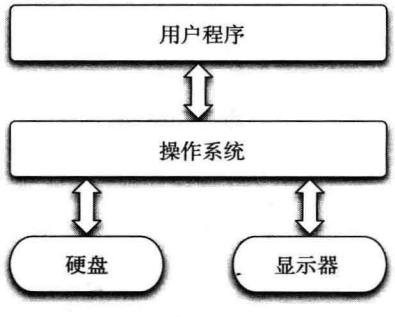


图 2-1 与操作系统通信

这看上去貌似有些多此一举，但是这也意味着只有编写操作系统的人才需要去为诸如“访问不同网卡”这样的事情而操心。其他人（比如分析科学数据的人或是制作 3D 虚拟聊天室的人）只需要明白如何利用操作系统即可令他们的程序运行于成千上万种不同的硬件环境了。

这就是 25 年前大部分程序员的工作方式。然而，现在的情况不同了，程序员和计算机硬件之间通常都会多出另外一个层次的东西。如果你的程序是用 Python、Java 或是 Visual Basic 编写的话，它们并不直接运行于操作系统之上，而是运行于一种称为解释器或虚拟机的程序之上。这种特殊的程序托管你的程序并运行，同时将你的命令翻译为操作系统所知的某种语言。与编写直接运行于操作系统之上的程序相比，这样做会更简单、更安全，而且更容易在不同操作系统之间进行移植。

不过，单单一个解释器还不够，还需要一个能够与真实世界交互的方式。一个办法是运行一种叫做壳（shell）的字符型程序，它可以读取来自键盘的命令，完成指定的任务，然后将结果以文本的形式显示出来，所有这一切都在一个窗口中完成。许多编程语言都有自己的 shell，操作系统也都有一个专门用来与之进行交互的 shell。本章我们将通过 Python 的 shell 程序来讲解 Python（见图 2-2）。

更为现代化的与 Python 进行交互的方式是使用集成开发环境（Integrated Development Environment，IDE）。这是一种成熟的图形界面，有着各式各样的菜单和窗口，就好像是 Web 浏览器、文字处理软件和绘图软件那样。

在教授如何编写程序时，我们最喜欢的 IDE 是免费的 Wing 101，它是一款轻量级的专业开发工具[⊖]。

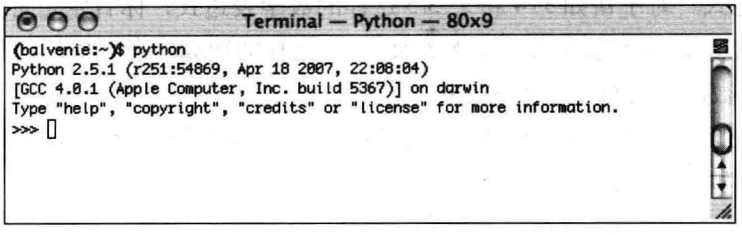


图 2-2　一款 Python 的 shell

另一款不错的 IDE 是 Python 自带的 IDLE。我们更倾向于使用 Wing 101，因为它是专门为初学者设计的，而 IDLE 则是一款较为复杂的开发环境。

Wing 101 的界面如图 2-3 所示。上半部分是编辑面板，我们将在这里编写 Python 程序；下面选项卡中标为“Python Shell”的部分是我们将要用来实验某些 Python 代码片段的地方。当学习到第 4 章时，我们将会更多地用到上面那个面板；目前我们将只关注 shell 这一部分。

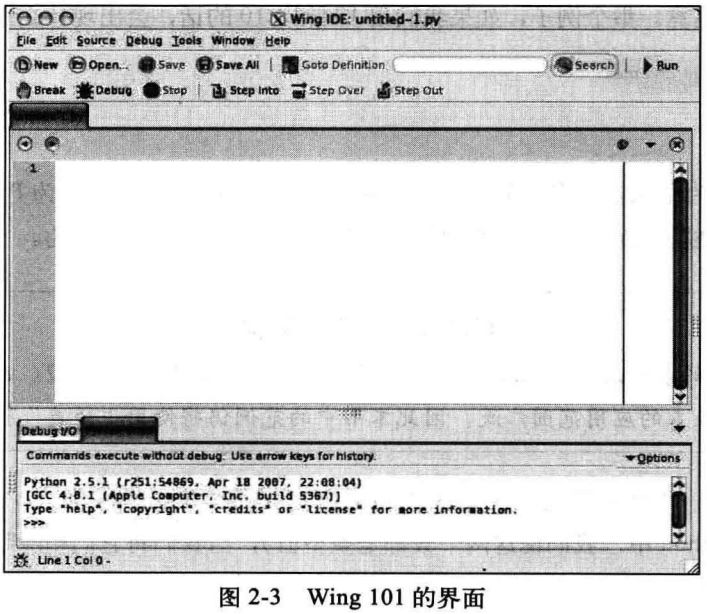


图 2-3　Wing 101 的界面

⊖ 更多详情请浏览 http://www.wingware.com。