



# 编程 Python

## 从入门到实战的16堂课

第2版

何敏煌 编著

**活用Python 3 程序设计，构建、开发动态网站**

快速介绍Python语言的精要，以实用为主的程序设计教学内容  
大量有趣的应用范例，通过做中学的方式强化学习成效  
善用Python模块，掌握以简短的程序代码做最佳的应用



清华大学出版社



编程

# Python

RFID

从入门到实战的16堂课

何敏煌 编著

第2版

清华大学出版社  
北京

## 内 容 简 介

本书第 2 版延续了第 1 版的风格，避开了难懂的程序设计语法，从介绍语言的精要入手，以活用 Python 3.x 实用的功能为核心内容，帮助读者达到熟练使用 Python 开发各种网络应用以及开发、制作和部署自己的动态网站的目标。

全书内容共分 16 章，第 1~4 章介绍程序设计基础知识，包括认识 Python 3 环境、程序包的管理与应用等；第 5~8 章以精心设计的示例讲授 Python 语言必备的基本语法、数据类型以及如何操作文件和数据库；第 9~13 章精选生活化的案例引导读者如何用学到的技巧以及网上资源提取和分析网页、操作在线实时数据库、处理图像文件等；第 14~16 章以一个动态网站的设计为主轴，学习 Django 2.0 网站的开发与部署、Scrapy 网络爬虫实践、嵌入式开发板的基本应用。

本书以“边练边学”的方式，从简单的程序入手，到掌握使用 Python 制作实用的动态网站，强化学习的成效，培养读者程序设计“实战”能力。本书既适用于教学和培训，又适合读者自学。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目 (CIP) 数据

Python 编程从入门到实战的 16 堂课/何敏煌编著. — 2 版. —北京：清华大学出版社，2019  
ISBN 978-7-302-52943-9

I. ①P… II. ①何… 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 083575 号

责任编辑：夏毓彦  
封面设计：王翔  
责任校对：闫秀华  
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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印装者：清华大学印刷厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：27.25

字 数：698 千字

版 次：2017 年 1 月第 1 版

2019 年 6 月第 2 版

印 次：2019 年 6 月第 1 次印刷

定 价：79.00 元

产品编号：081828-01

# 前 言

一如大家的期待，Python 持续以飞快的速度不断地发展完善。非常感谢广大读者和院校老师们对本书第 1 版的厚爱，然而，随着时间的流逝和技术的发展，我们增加的不只是岁月在脸上留下的皱纹以及头上的白发，还有一大堆程序包以及程序包的版本号，版本号的增加代表的通常是更多的功能以及更稳定好用的程序包，这是件好事，但对于学习者来说，象征着更多的改变要去适应和学习，当然对于作者来说，则是改版和更新版本的动力来源。

为了能够让读者和同学们可以更快地适应 Python 生态环境的更新换代，作者对本书做了大幅的更新和修改，除了全部采用 Python 3 作为程序设计的基准、调整了一些相关的软件操作新版插图之外，Anaconda 成为重点，因为 Anaconda 已经成熟到在各个操作系统中安装几乎都不会出现任何问题，在 Windows 10 的环境下也运行得很好，所以本书的示范过程多是在 Anaconda Prompt for Windows 10 的界面中完成的，这样让读者和同学们可以更容易地练习。建议打算学习 Python 的读者，不用多想，先去把 Anaconda 安装好再说。

除了本书第 1 版原有的内容之外，在第 4 章特别加上好玩的 Google 语音模块，在短短的几行程序代码中就可以使用程序控制 Google 小姐说话，让我们的程序在应用上更为有趣。第 9 章和第 10 章花了许多篇幅讲述在最新的网页进行数据提取的程序及自动化提取技术，第 11 章则针对 Firebase 做了大幅的更改，在接口的程序部分做了许多的修改，使用了另一个比较新的程序包。

此外，在大数据挂帅的今日，除了原有的 MySQL 数据库服务器之外，NoSQL 是近年来兴起的非常热门的数据库类型，作者特别在第 12 章中追加了 MongoDB 服务器操作的教学，让读者在存储大量数据的时候更加地顺手。

受到大家喜爱的 Django Web 框架已经进展到 2.0 版了，因此第 14 章做了非常大的修改。此外，第 15 章有关部署网站的部分，我们改用 Pythonanywhere 这个好用的云计算接口，另外 Heroku 的部署也变得非常简单。最后，在第 16 章除了原有的内容之外，在这一版中增加了 Scrapy 网络爬虫框架的应用，相信大部分同学一定会喜欢。现在非常热门的嵌入式开发板 BCC micro:bit 也可以使用 Python 语言开发程序了，在本书的最后一节跟上了这股潮流，协助有兴趣的读者更快上手。

最后，感谢读者和授课老师们百忙中来信指正第 1 版的许多排版错误，也感谢帮助测试本书大部分的网站以及文字校稿的何旻同学，这些努力让本书的内容进一步完善。不过，虽然作者尽力确保书中每一个教学步骤与内容的正确性，但是有可能“百密一疏”，敬请读者、同学和老师们继续提供宝贵意见与指正。

何敏煌  
2019 年 4 月

# 改编说明

自从 2004 年以来，Python 程序设计语言的使用率一直呈线性增长，毫无疑问，它已经成为最受欢迎的程序设计语言之一。作为一款纯粹以自由软件方式推广的程序设计语言，Python 的语法简洁清晰，并且可以把丰富和强大的链接库（包含其他语言制作的各种模块）很轻松地链接在一起，所以它又有“胶水”语言的美誉。因为简单易用而且功能强大，所以不仅仅是专业人员在用，而且越来越多的计算机用户也开始使用 Python 提高自己运用计算机的能力。

本书第 2 版延续了第 1 版的风格，与传统的教授程序设计语言的教材相比，本书的结构与叙述风格更加“亲民”。为了避免读者在学习程序设计语言的过程中出现常见的从“望而却步”到“勉为其难”，再到最终“弃学”的窘境，本书从一开始就绕开了“枯燥乏味”的程序设计语言的语法和程序设计过程要注意的“琐碎”事项，更没有把重心放在展示程序设计技巧方面。纵观全书，各个章节都是以精选的日常问题为主线，让读者分析和学习这些日常问题的解决方法，在饶有兴趣的“实战”中轻轻松松就学会了运用强大的 Python 语言来“解决”实际问题。

一些知名大学已经采用 Python 语言来教授计算机程序设计课程了，本书有助于学生拓展自己的程序设计“实战”能力。而对于有意转向使用 Python 语言来开发网络应用，甚至是开发和部署完整的网站系统的专业人员来说，本书可以作为学习 Python 路途中的“导航仪”。因此，本书既适合于教学和培训，也适合读者自学。

本书提供的范例程序可扫描以下二维码获得：



如果下载有问题，请发送电子邮件至 [booksaga@126.com](mailto:booksaga@126.com)，邮件主题为“Python 编程从入门到实战的 16 堂课（第 2 版）”。

这些程序源代码读者需要根据自己的具体运行环境进行相应的修改，并不是所有的范例程序拿来就可以直接运行。

最后祝大家学习顺利，早日成为 Python 领域的技术“大腕”！

资深架构师 赵军  
2019 年 5 月

# 目 录

第 1 章 程序设计所需要的基础知识 .....	1
1-1 什么是程序设计语言 .....	1
1-2 程序设计的重要性 .....	3
1-3 最受欢迎的程序设计语言 .....	4
1-4 学习程序设计需要知道的逻辑概念 .....	6
1-5 本书的结构及内容说明 .....	8
第 2 章 快速了解 Python 程序设计语言 .....	10
2-1 Python 简介 .....	10
2-1-1 Python 的历史沿革 .....	10
2-1-2 Python 的重要性 .....	11
2-1-3 Python 程序设计基本元素 .....	12
2-1-4 Python 程序易用性示范 .....	13
2-2 学习 Python 的重要性 .....	16
2-3 Python 2 和 Python 3 的差异 .....	16
2-4 Python 的应用领域 .....	17
2-5 习题 .....	18
第 3 章 建立可以开始编写程序的 Python 环境 .....	19
3-1 马上使用 Python 编写程序 .....	19
3-2 安装 Python 3.x 窗口环境 .....	23
3-2-1 Windows 的 IDLE 窗口环境 .....	23
3-2-2 Microsoft Visual Studio 的 Python 开发环境 .....	26
3-2-3 Anaconda 的安装与使用 .....	28
3-3 简单且易上手的 IPython Notebook 和 jupyter .....	30
3-4 程序代码编辑器的介绍 .....	36
3-4-1 Notepad++ 的安装与应用 .....	36
3-4-2 TextWrangler 的安装与应用 .....	40
3-5 在 Linux 虚拟机中运行 Python .....	42
3-6 习题 .....	52
第 4 章 Python 程序包管理与在线资源 .....	53
4-1 Python 程序包管理工具 .....	53
4-1-1 easy_install 的安装与使用 .....	54
4-1-2 pip 的安装与使用 .....	54
4-2 Python 虚拟环境的设置 .....	55
4-2-1 在 Mac OS 中安装 virtualenv .....	55

4-2-2	在 Windows 中安装 virtualenv	56
4-3	高级程序包安装实践	57
4-3-1	conda 程序包管理程序的使用	58
4-3-2	使用 Matplotlib 绘制精美数学图形	59
4-4	Python 的在线资源与支持	61
4-4-1	搜索 PyPI 相关信息的方法	61
4-4-2	产生数独题目的程序包的应用	63
4-4-3	Google 文字转语音程序包的应用	65
4-4-4	寻求在线支持	68
4-5	习题	68
<b>第 5 章</b>	<b>开始设计 Python 程序</b>	<b>69</b>
5-1	jupyter 的介绍与使用	69
5-1-1	IPython	69
5-1-2	在 Windows 操作系统中变更 IPython 的默认编辑器	70
5-1-3	jupyter notebook 的操作	72
5-2	程序的构想与实现	76
5-2-1	理清问题的需求	76
5-2-2	数据结构	77
5-2-3	算法与流程图	78
5-2-4	开始设计程序	80
5-2-5	调试	83
5-3	猜数字游戏	84
5-3-1	问题需求	84
5-3-2	数据结构	84
5-3-3	算法与流程图	84
5-3-4	完成程序	86
5-4	习题	87
<b>第 6 章</b>	<b>Python 简易数据结构速览</b>	<b>88</b>
6-1	常数、变量和数据类型	88
6-1-1	常数和变量的差异	88
6-1-2	变量的命名原则	90
6-1-3	保留字	90
6-1-4	基本数据类型	91
6-2	Python 表达式	94
6-2-1	基本表达式	94
6-2-2	关系表达式	95
6-2-3	逻辑表达式	96
6-3	列表、元组、字典和集合	97
6-3-1	列表与元组	97

6-3-2	列表的操作应用	100
6-3-3	字典 dict	103
6-3-4	集合 set	105
6-3-5	查看两个变量是否为同一个内存地址	106
6-4	内建函数和自定义函数	108
6-4-1	内建函数	108
6-4-2	自定义函数	110
6-4-3	import 与自定义模块	113
6-5	单词出现频率的统计程序	115
6-6	习题	116
<b>第 7 章</b>	<b>程序控制流程</b>	<b>117</b>
7-1	判断语句的应用	117
7-2	循环语句	120
7-3	高级循环指令	123
7-4	例外处理	126
7-5	程序流程控制的应用	129
7-6	习题	135
<b>第 8 章</b>	<b>文件、数据文件与数据库的操作</b>	<b>136</b>
8-1	文件与目录的操作	136
8-1-1	os.path	136
8-1-2	glob	138
8-1-3	os.walk	139
8-1-4	os.system 和 shutil	140
8-2	数据文件的操作	142
8-2-1	文本文件的读取与写入	142
8-2-2	文本文件的应用	148
8-2-3	读取 JSON 格式的数据	151
8-3	Python 与数据库	155
8-3-1	安装 Firefox 的 SQLite Manager 附加组件	155
8-3-2	创建简易数据库	158
8-3-3	Python 存取数据库的方法	160
8-4	数据库应用程序	161
8-5	习题	167
<b>第 9 章</b>	<b>用 Python 自动提取网站数据</b>	<b>168</b>
9-1	因特网程序设计基础	168
9-1-1	因特网与 URL	169
9-1-2	解析网址	172
9-1-3	提取网页数据	175

9-1-4	提取网页内的电子邮件账号	177
9-2	网页分析与应用	179
9-2-1	HTML 网页格式简介	179
9-2-2	安装 Beautiful Soup	183
9-2-3	使用 Beautiful Soup 提取信息	184
9-2-4	进一步分析网页的内容	188
9-3	网络应用程序	190
9-3-1	将数据存储为文件	191
9-3-2	以网页的形式整理数据	193
9-3-3	在本地建立网页应用	196
9-4	习题	198
<b>第 10 章</b>	<b>Python 网页数据提取实践</b>	<b>199</b>
10-1	把网页数据存储到数据库中	199
10-1-1	网页数据的运用模式	200
10-1-2	把数据存储到 SQLite	201
10-1-3	把数据导入网络 MySQL 数据库中	206
10-1-4	编写本地程序读取网络 MySQL 数据库中的数据	210
10-1-5	使用 PHP 建立信息提供网站	211
10-2	自动提取数据	214
10-2-1	检测网页内容是否曾经更新	214
10-2-2	Windows 自动化设置	219
10-2-3	Mac OS 自动化设置	223
10-3	通过 Python 操作浏览器	223
10-3-1	安装 Selenium	224
10-3-2	使用 Selenium 操作 Chrome	226
10-3-3	通过 Selenium 读取网页信息	228
10-3-4	登录会员网站的方法	230
10-4	习题	233
<b>第 11 章</b>	<b>Firebase 在线实时数据库操作实践</b>	<b>234</b>
11-1	Firebase 数据库简介	234
11-1-1	NoSQL 数据库概念	235
11-1-2	注册 Firebase 账号	235
11-1-3	连接 Firebase 和 Python	241
11-2	Python 存取 Firebase 数据库的实例	243
11-2-1	Firebase 网络数据库的操作	243
11-2-2	使用 Python 写入 Firebase 数据库	244
11-2-3	使用 Python 读取 Firebase 数据库	246
11-2-4	整合范例	249
11-3	网页连接 Firebase 数据库	253

11-3-1	Firebase Hosting 免费主机空间的设置	253
11-3-2	使用 JavaScript 读取 Firebase 数据库	258
11-3-3	Firebase 网页设计	259
11-4	Firebase 数据库的安全验证	261
11-4-1	Firebase 安全性的设置	262
11-4-2	电子邮件地址/密码的登录方式	263
11-4-3	Python 端的设置	265
11-4-4	将具有用户验证功能的数据写入程序	267
11-5	习题	268
<b>第 12 章</b>	<b>Python 应用实例</b>	<b>269</b>
12-1	Facebook Graph API 的介绍与使用	269
12-1-1	安装 facebook-sdk	269
12-1-2	Facebook Graph 简介	270
12-1-3	Python 程序存取 Facebook 设置	275
12-1-4	通过 Python “发表”文章	277
12-1-5	下载在 Facebook 中的照片	279
12-2	照片文件的管理	281
12-2-1	照片文件的分析	281
12-2-2	找出重复的照片文件	283
12-2-3	将照片文件重新编号	286
12-3	找出网络中最常被使用的中文词	287
12-3-1	搜索新闻文章	287
12-3-2	安装中文分词模块 jieba	288
12-3-3	找出文章中最常被使用的词汇	289
12-4	MongoDB 数据库操作实践	291
12-4-1	建立本地的 MongoDB 数据库	291
12-4-2	使用 Python 操作 MongoDB 数据库	296
12-4-3	MongoDB 数据库应用实例	299
12-5	习题	300
<b>第 13 章</b>	<b>Python 绘图与图像处理</b>	<b>301</b>
13-1	Matplotlib 的安装与使用	301
13-1-1	Matplotlib 介绍	301
13-1-2	使用 Matplotlib 画图	303
13-1-3	统计图的绘制	306
13-1-4	数学函数图形的绘制	312
13-2	pillow 的安装与使用	316
13-2-1	pillow 简介	316
13-2-2	读取图像文件的信息	317
13-2-3	简易图像文件处理	318

13-3	批量处理图像文件	321
13-3-1	为自己的照片加上专属标志和批量调整照片尺寸	321
13-3-2	中文字体的处理与应用	323
13-3-3	为图像文件加入水印功能	327
13-4	习题	329
<b>第 14 章</b>	<b>用 Python 打造特色网站</b>	<b>330</b>
14-1	使用 Python 编写一个网站程序	330
14-1-1	网站原理	330
14-1-2	网站程序的输入与输出	331
14-1-3	使用 Python 编写的网站框架	332
14-2	Django 简介	335
14-2-1	下载与安装 Django	335
14-2-2	Django 目录及重要配置文件解说	338
14-2-3	前端与后端的搭配	340
14-2-4	建立你的第一个 Django 网站	341
14-3	认识 Django Framework 的架构	342
14-3-1	Django 的 MTV 架构	342
14-3-2	URL 的对应方法详解	343
14-3-3	模板的使用	344
14-3-4	使用静态文件夹存取文件	348
14-4	Django 与数据库	348
14-4-1	在 Django 中使用数据库	349
14-4-2	建立模型	350
14-4-3	admin 后台管理	351
14-4-4	读取数据库中的数据	354
14-4-5	短网址转址网站模板的内容	356
14-5	习题	359
<b>第 15 章</b>	<b>Django 网站开发与部署</b>	<b>360</b>
15-1	网站的测试与调整	360
15-1-1	上线前的前置工作	360
15-1-2	网站的部署策略	361
15-1-3	网址的购买和选用	362
15-2	网站开发环境的部署	362
15-2-1	ngrok	363
15-2-2	申请 pythonanywhere 账号	364
15-2-3	建立 pythonanywhere 网站开发环境	366
15-2-4	测试与执行 Django 网站	370
15-3	云计算虚拟机部署方法	374
15-3-1	DigitalOcean 简介	374

---

15-3-2	创建 Ubuntu 虚拟机	376
15-3-3	安装、设置 Apache 服务器和 Django Framework	376
15-3-4	上传文件和网站上线	377
15-4	云计算 App 主机部署	381
15-4-1	Heroku 简介	381
15-4-2	创建 Heroku 账号	382
15-4-3	在 Windows 10 操作系统中部署 Heroku	383
15-4-4	在 Heroku 上部署 Django 网站	384
15-5	习题	387
<b>第 16 章</b>	<b>提升 Python 能力的下一步</b>	<b>388</b>
16-1	程序代码的版本控制	388
16-1-1	Git 简介	389
16-1-2	Git 实践操作	392
16-1-3	BitBucket 的申请使用	399
16-2	Scrapy 网络爬虫框架应用实例	404
16-2-1	Scrapy 的安装	404
16-2-2	简易爬虫程序的实现	406
16-2-3	爬虫程序与数据库的整合	413
16-3	嵌入式系统与 Python	416
16-3-1	BBC micro:bit 简介	416
16-3-2	使用浏览器设计 micro:bit 程序	417
16-3-3	使用 Mu Editor 设计 micro:bit 程序	420
16-4	提升学习的下一步	422

# 第 1 章

## 程序设计所需要的基础知识

- ※ 1-1 什么是程序设计语言
- ※ 1-2 程序设计的重要性
- ※ 1-3 最受欢迎的程序设计语言
- ※ 1-4 学习程序设计需要知道的逻辑概念
- ※ 1-5 本书的结构及内容说明

### 1-1 什么是程序设计语言

人和人之间的沟通需要语言，人和计算机沟通当然也需要使用语言，只不过现在计算机技术还没有进步到可以完全听懂人类使用的语言，因此想要让计算机帮我们做事情，只好发明一种比较严谨、语法限制比较多但是比较容易让计算机“理解”的语言，这一类语言统称为计算机语言。

就像不同国家、民族的人讲话会使用各种各样的语言和语法，与计算机沟通用的语言随着应用环境和计算机设备不同以及当初设计计算机语言的工程师（发明人）想法的不同，也有许多不同种类的语法格式，有些陈述方式比较类似，有些则非常不一样，各有各的名字和用途以及长处和短处，这也是为什么没有一种全世界统一的计算机语言的原因。各种不同的计算机语言活跃在各自的领域，所以，程序设计语言就如同人类的语言一样，也有非常多的种类可以选择，常见的计算机语言（因为可以用来编写程序，所以又被称为程序设计语言）有 Assembly、ASP、BASIC、C、C++、C#、Java、JavaScript、Pascal、Python、Ruby、Forth、Perl、PHP 等，前后在不同年代出现过上百种。所有曾经出现过的程序设计语言在维基百科中做了整理：

[https://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_programming_languages)

虽说是计算机语言，但是却不像人类那样可以用讲话的方式说给计算机听（也有，但是不成熟，并未到实用的阶段），这需要其他技术来实现，而且对于要求高效率执行的程序来说，用说话的方式并不符合实际的用途，所以要让计算机执行某些我们要求的工作，必须用写的语言，也就是我们说的“程序”。“程序”可以看作是一个“脚本”，或者一张（工作复杂的话，也可能会有好多张）写满了要计算机工作的任务列表，当计算机收到这个“脚本”的时候，会按照上面的指示一项一项地把任务做完，如图 1-1 所示。

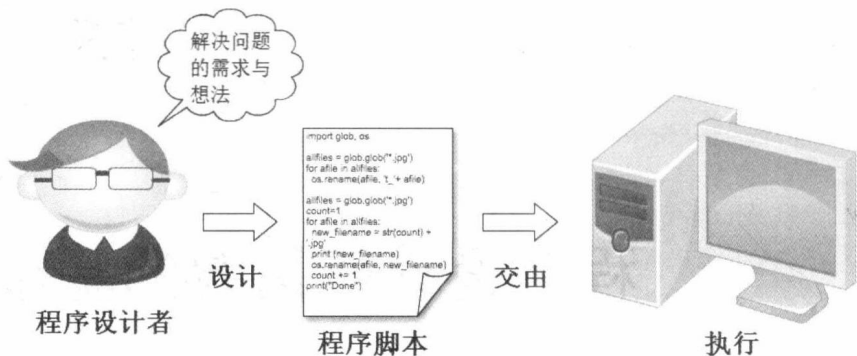


图 1-1 用计算机语言写成的脚本要交给计算机去执行

可以想象成，计算机就是一堆组合在一起具有工作能力的电子元器件和电路板（统称为“硬件”，其中最重要的是 CPU）。如果没有特别的指示和要求，它们并不会主动地解决任何问题，所有的行为都需要人们（或更精确一点说，懂得编写程序的人）把所有要计算机做的事项写在一些文件中。在计算机开机的时候读取这些文件，照着文件上的指示执行特定的工作，而这些可以执行的文件里面存放的就是之前编写程序的人写出来的程序代码脚本，再经过一层层的翻译之后，就成为可以在计算机的中央处理单元（CPU）中执行的机器语言指令集合。

从微观的角度来看，计算机中所有部件的运行都需要不同层级的程序，每一件事都要通过计算机工程师所编写的程序去执行。然而，对于初学者来说，如果每一件事都要亲力亲为，那么只有非常厉害的工程师才有足够的能力使用计算机。所幸，大部分底层的工作都已由计算机工程师解决了，计算机用户接触到的层级已经到了 Windows 10/Mac OS X 这一类高级图形化操作系统以及 Chrome、Edge、Microsoft Office 这一类应用程序，只要使用鼠标和键盘，就可以开始工作（或进行游戏）了。

如今人们桌面上的个人计算机都属于通用型计算机（General Purpose Computer），意思是计算机本身没有特定的应用目的，就是提供计算能力以及硬件资源给用户使用，能够解决什么问题取决于用户执行了什么应用程序：执行了浏览器就可以上网，执行了游戏软件就可以娱乐休闲，执行了会计软件就可以协助处理会计事务，而执行了统计软件则可以协助处理大量的统计数据以及绘出分析结果等。这些程序和应用软件都是计算机工程师辛苦编写出来的，从而实现特定的目标。

有别于通用型的计算机，在任何机器或设备中（包括在飞机、高铁以及汽车上）使用的计算机都属于特定目的型的计算机（Special Purpose Computer）。设计它们的目的是为了完成某一项特定的工作，在大部分情况下，一般用户是接触不到这种计算机的程序设计层面的。

想要学习程序设计的朋友，也可以直接从高级的程序设计语言（比较接近人类思考模式的程

序设计语言)入手,在程序开发用的集成开发环境或可以处理文字的程序编辑器中,把要计算机做的工作事项以特定程序设计的语句和语法写出来,然后会有一个负责翻译的程序(程序设计的编译器或者解释器)把这个写好的程序脚本翻译成计算机看得懂的格式让计算机去执行,如图 1-2 所示。

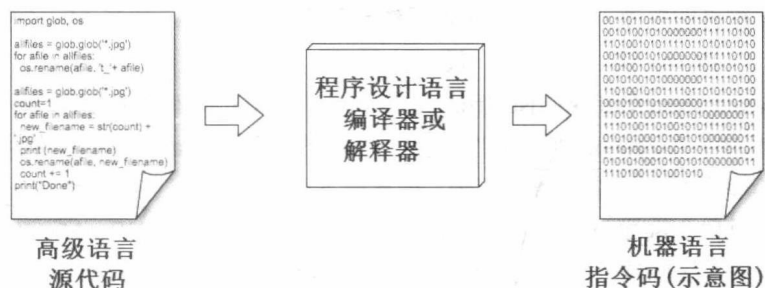


图 1-2 程序要经过翻译才能够被计算机执行

这些程序设计语言要写在哪里才有编译器或者解释器可以协助翻译并让计算机执行呢?传统的程序设计语言(如 Basic、C/C++等)因为需要翻译以及交付计算机执行程序的操作多而且复杂,所以要安装特定公司开发的程序设计语言开发环境(如 Microsoft 公司的 Visual Studio)才行,例如想要编写 Java 程序,就要有 JDK 以及设置好的开发环境,如 Eclipse 等。不过,现在情况已经不同了,进入网络时代之后,可选的开发方式多了不少,像 JavaScript 就是一个在浏览器中执行的程序设计语言,几乎所有的图形化操作系统(Windows、Mac OS、Linux 的 X Window)都提供了浏览器,我们只要使用文本编辑器(记事本这一类小编辑器都可以)写好 JavaScript 的脚本,就可以通过浏览器(Internet Explorer、Chrome、Firefox、Safari)来加载执行,省去要先安装搭建程序执行环境的困扰。如果编写好的程序所运行的环境没有特别的要求,单纯只是要进行运算并显示出结果,或者要以网页来作为输出的界面,那么有许多在线的编译器可以让编写好的程序直接在网页上运行,例如 repl.it、Cloud9、JSFiddle、Ideone、CodeGround 等。

和其他传统程序设计语言相比,本书的主角 Python 直接提供了交互式的界面,只要安装之后,就可以在它的交互式文字界面中执行以及编写程序。如果你的计算机中执行的操作系统是 Mac OS 或 Linux,那么连安装都不需要,操作系统默认就内建了 Python 解释器以及所有相关的模块。在 Mac OS 和 Linux 中,很多好用的工具程序都是使用 Python 编写成的。因此,如果你的个人计算机操作系统使用的是 Mac OS 或 Linux (CentOS、Ubuntu、Fedora 等),那么不用考虑安装的问题,直接在命令行(终端程序 Terminal)下输入 python 或 python3 (后者是执行第 3 版的 Python 解释器),随后即可使用 Python 程序设计语言来设计程序;如果你的操作系统是 Windows 系列,就需要一些安装步骤(当然,如果你使用的是 repl.it 这一类在线程序设计学习环境,就不需要安装),在后面的章节中会详细介绍安装过程。

## 1-2 程序设计的重要性

简单地说,程序设计就是把想要解决的问题加以详细地分析,把要处理的数据抽象化,然后将这些数据转化为计算机中的一些代码存储起来,再根据解决此问题的步骤一步一步地针对这些代

码进行必要的运算,最后输出结果。只要分析得当,几乎所有分析过的问题都能够被加以处理和解决。最重要的是,因为计算机的计算能力非常强,而且可以每天 24 小时不间断地运行而不会有任何的怨言,等于是只要设计部署得当,计算机可以随时为我们不间断地工作。这项特点在网络时代的今日显得更加重要。

想象一下,计算机的电源一旦开启,操作系统(无论是 Windows 还是 Mac OS)本身就是一个庞大而复杂、由一大堆程序代码所组成的系统程序组,根据网站 <http://www.informationisbeautiful.net/visualizations/million-lines-of-code/> 显示的数据,Windows 7 用了约 4000 万行程序代码,Facebook 则用了约 6100 万行程序代码,Mac OS 10.4 版用了约 8500 万行程序代码,而 Google 的所有网络服务加起来,推算一下可能超过 20 亿行程序代码。在操作系统启动完成之后,一般用户会执行 Office、Photoshop、Acrobat Reader、LINE、Movie Maker 或浏览器来处理工作上的业务,这些软件都是程序设计师辛苦工作的成果,有了这些系统软件及应用程序,用户只要运用鼠标和键盘就可以开始日常的工作了。

因此,要成为一位程序设计师或计算机专业人员,程序设计能力是一项非常重要的专业技能。但是,对于一般不是以计算机为专业的用户来说,程序设计重要吗?这个问题在 Python 这一类快速弹性化的程序设计语言还没有出现之前,回答或许是否定的,但是功能强大且易上手的 Python 问世之后,这个答案是 100%肯定的。原因在于程序执行的精髓:个性化和自动化。

使用现有的应用程序可以迅速地以鼠标和键盘来操作,以实现用户的想法,但是许多工作或项目其实隐含着高度的重复性和时间相关性。举例来说,在学生毕业季来临的时候,为了找到心仪的工作,需要每日搜索各大招聘网站和各大公司招募人才的广告,或者正在关注股票投资信息的散户,想要在国内或欧美股市收盘时立刻汇集整理特定类型的股票或个股的相关成交信息并加以分析,这些工作如果以人工来做的话,工作情况如何呢?除非已购买或已请程序设计师设计编写了客户化的相关程序,否则需要用户自行在特定的时间通过浏览器去各个相关网站搜索和查看数据,然后把这些数据复制到 Word 或 Excel 等程序中加以整理分析。这样不仅操作重复且步骤烦琐,而且人工执行这些工作时也容易出现疏漏。此外,更重要的是,新闻网站会实时更新,而且欧美股市均在北京时间的深夜和凌晨时才开盘或者收盘,人工操作不只是精确度不佳,而且人也太过劳累。

熟悉计算机系统的用户(强力用户,Power User)可以通过操作系统的各种设置达成自动化执行某些程序的目的,但是如果熟悉程序设计的话,这些工作都可以通过适当的程序代码以自动化方式来完成。在 Python 出现之前,这样的程序解决方案不是没有,但是设计起来都非常复杂,不适用于一般的计算机用户;在 Python 出现之后,不需要厉害的程序设计师,即使是一般的计算机用户,也可以通过简短的 Python 程序实现上述目标。这也是作者编写本书推广 Python 语言最大的原因——非专业的计算机用户也可以通过简短的程序代码让计算机具有更佳的自动化能力,从而提升计算机用户的工作效率。

## 1-3 最受欢迎的程序设计语言

尽管本书的目的是介绍 Python 程序设计,不过对于程序设计的初学者来说,也有必要知道如今在计算机界中非常活跃的其他程序设计语言。2015 年最受欢迎的程序设计语言,一个比较学术

上的统计（数据源 IEEE，网页为 <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>）如下：

1. Java
2. C
3. C++
4. Python
5. C#
6. R
7. PHP
8. JavaScript
9. Ruby
10. Matlab

另一份则是来自 CodeEval（参考网址为 <http://blog.codeeval.com/codeevalblog/2015>）：

1. Python
2. Java
3. C++
4. C#
5. Ruby
6. JavaScript
7. C
8. PHP
9. Go
10. Perl

C 语言是非常经典的程序设计语言，现今大部分的操作系统底层仍是以 C 语言来编写的，而且其他许多语言（如 C++、C#、Java、PHP）也可以看成是 C 语言衍生出来的，因为承袭了相当多的 C 语言元素和设计精神，如果想要成为程序设计师，C 语言的学习是不可或缺的。

在这份列表中，传统的以教学为目的的程序设计语言 BASIC 和 Pascal 早已不见踪影，BASIC 还可以在微软的集成式开发工具 Visual Studio 中看到，在 Microsoft Office 的宏指令设计中也是 Visual Basic 的发挥场所，同时在 ASP 中也有 Visual Basic 的影子，但是 Pascal 已完全看不到应用的地方了。

PHP 是属于网页后端的程序设计语言，几乎都是在网页服务器中执行，PHP 这么受欢迎的原因，有很大一部分是因为许多有名的 CMS 系统（如 WordPress、Joomla、Drupal、OpenCart）现在还是由 PHP 建构而成，不过有一部分系统（如 WordPress）现在也有了转换程序设计语言的预兆，所以 PHP 是否能够持续占据此排行榜就难以预料了。而 JavaScript 则是刚好相反，它是在浏览器中执行的语言，有 Google 公司的 AngularJS 这个 Framework 的支持，以及 NodeJS 后端 Framework 和教育用嵌入式 micro:bit 的蹿红，在未来的数年内稳定地留在榜内是绝对没有问题的。

Ruby 和 Python 这两种程序设计语言目前在网络上都拥有非常活跃的社区和拥护者，不只在前