基于最新的Python 3.x版本进行讲解，与2.x有区别的地方附有相关说明，适合2.x和3.x的读者。

涵盖用Python操作数据库、进行Web开发、网络编程、科学计算、多线程编程等高端领域。

书中提供三大案例，分别涉及用Python进行Windows优化、大数据处理和游戏开发。

杨佩璐　宋　强　等编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

宝 典 丛 书

Python 宝典

杨佩璐　宋强　等编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

Python 是目前流行的脚本语言之一。本书由浅入深、循序渐进地为读者讲解了如何使用 Python 进行编程开发。全书内容共分三篇，分为入门篇、高级篇和案例篇。入门篇包括 Python 的认识和安装、开发工具简介、Python 基本语法、数据结构与算法、多媒体编程、系统应用、图像处理和 GUI 编程等内容。高级篇包括用 Python 操作数据库、进行 Web 开发、网络编程、科学计算、多线程编程等内容。案例篇选择了 3 个案例演示了 Python 在 Windows 系统优化、大数据处理和游戏开发方面的应用。

本书针对 Python 的常用扩展模块给出了详细的语法介绍，并且给出了典型案例，通过对本书的学习，读者能够很快地使用 Python 进行编程开发。

本书适合 Python 初学者、程序设计人员、编程爱好者、本科及大专院校学生，以及需要进行对科学的计算的工程人员阅读。

图书在版编目（CIP）数据

Python 宝典 / 杨佩璐等编著. —北京：电子工业出版社，2014.5
（宝典丛书）
ISBN 978-7-121-22562-8

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

中国版本图书馆 CIP 数据核字（2014）第 047661 号

策划编辑：张月萍
责任编辑：徐津平
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×1092　1/16　　印张：31.5　　字数：850千字
印　　次：2014年5月第1次印刷
定　　价：79.80元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888

前　言

Python 是一种功能强大的脚本语言。使用 Python 可以完成从文本处理到创建复杂的三维图形等各种工作。在企业级应用中，由于 Python 具有简洁的语法和丰富的扩展模块，因此，使用 Python 可以大幅缩短开发周期，节约成本。

另外，Jython 还可以在 Java 中使用 Python，通过 Python 的灵活性来提高 Java 在企业级应用的效率。在 Web 方面，有很多基于 Python 的流行 Web 框架，如 Zope/Plone、Django、TurboGear 等。通过这些 Web 框架，程序员可以使用 Python 快速构建安全、功能强大的网站。

在数值计算与工程应用方面，Python 与传统的 C 和 Fortran 相比，更加灵活、简洁，并且可以十分方便地创建 GUI 界面。通过使用 SciPy 模块和 Matplotlib 绘图库可以进行数值计算，实现工程数据的可视化。

如何学习本书

本书内容共分三篇 26 章，分别为入门篇、高级篇和案例篇。

入门篇为第 1 章至第 15 章，从初识 Python 开始，由浅入深地介绍了 Python 的安装、开发工具的使用、Python 数据类型与基本语句、可复用的函数与模块、数据结构与算法、面向对象的 Python、异常处理与程序调试、Python 多媒体编程、使用 PIL 处理图片、系统编程、使用 Python 的 GUI 编程等内容。其中，第 11 章至第 15 章专门对 Python 的几种 GUI 编程工具进行了讲解，读者可以比较各种 GUI 编程工具，根据自己的兴趣及以前所学的知识，选择适合自己使用的 GUI 编程工具。

高级篇为第 16 章至第 23 章，主要介绍了用 Python 操作数据库、进行 Web 开发、网络编程、科学计算、多线程编程等内容。其中，第 22 章介绍了 Python 的扩展和嵌入，需要读者有 C/C++ 的相关背景，如果读者不会使用 C/C++进行编程，可以跳过该章。在大部分开发场景中，不掌握这部分知识也不影响用 Python 编程。

案例篇为第 24 章至第 26 章，每章介绍一个案例，包括用 Python 优化 Windows、用 Python 处理大数据和用 Python 开发《植物大战僵尸》游戏。通过对这 3 个案例的学习，可进一步巩固读者前面所学的知识。

本书以 Python 3.x 为基础进行讲解，并在与 Python 2.x 有区别的地方加上了相关介绍，使 Python 2.x 和 Python 3.x 的读者都能使用本书。

本书特色

- 内容全面，对 Python 各方面的知识都做了系统详尽的讲解。

- 结构清晰，全书整体结构上遵循从易到难、由浅入深的顺序。
- 内容新颖，结合当前最新的 Python 3.x 版本进行讲解。
- 实用性强，本书在各章节中都有大量程序示例，并在最后一篇详细讲解了 3 个不同方向的开发实例。
- 实例丰富，对于每一个知识点，书中都通过相应的示例进行讲解。

适合的读者

- Python 初学者
- 程序设计人员
- 编程爱好者
- 本科及大专院校学生
- 需要进行科学计算的工程人员

本书由杨佩璐（山东中医药大学）、宋强（安阳工学院）共同编写，其中杨佩璐编写了本书的第 1~13 章，宋强编写了本书的第 14~20 章，同时参与编写的还有徐新其、张俊华、赵桂芹、张增强、刘桂珍、李杰、曾智海、代得新、邵星星、过建军、王正江、朱林鑫、张伟杰、田亮亮，在此一并表示感谢。

在本书的编写过程中，编者竭尽全力，不敢有丝毫疏忽，并对所有程序都进行了上机调试，但由于 Python 发展非常迅速，仍会有不同版本的差别，因此需要读者多加注意。另外，书中难免有不足之处，望广大读者批评指正。

编者

2014 年 1 月

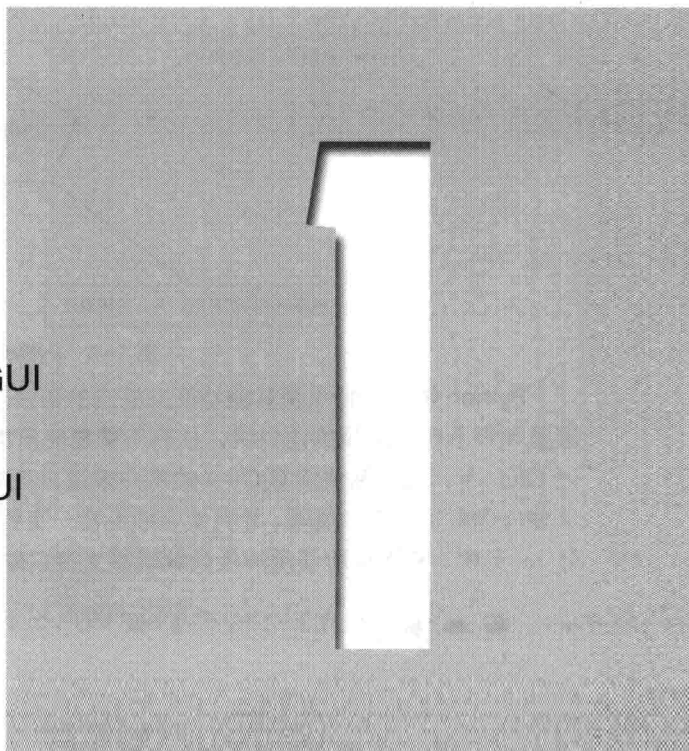
目　录

第 1 部分　入门篇

第 3 部分 案例篇

Part

第 1 部分　入门篇



第 1 章　初识 Python

本章包括

- 什么是 Python
- 其他程序设计语言中的 Python
- 用 VS2008 编译 Python 源码
- 第一个 Python 程序
- Python 的优点
- Python 的下载和安装
- Python 开发工具：Vim、Emacs 和 PythonWin

Python 是一种面向对象的、直译式计算机程序设计语言，也是一种功能强大且完善的通用型语言，已经具有二十多年的发展历史，成熟且稳定。Python 是免费的语言，具有面向对象的特性，可以运行在多种操作系统之上。Python 具有清晰的结构、简洁的语法以及强大的功能。Python 可以完成从文本处理到网络通信等各种工作。Python 自身已经提供了大量的模块来实现各种功能，除此之外，还可以使用 C/C++来扩展 Python，甚至还可以将 Python 嵌入到其他语言中。

1.1 Python 是什么

Python 是目前流行脚本语言之一，它是由 Guido van Rossum 创建的。Python 语言主要受到教学语言 ABC 和 Modula-3 的影响。Python 被设计得简洁、优美却又不失脚本的灵活性和拥有强大的功能。Python 的主要特点可以用图 1-1 来表示。

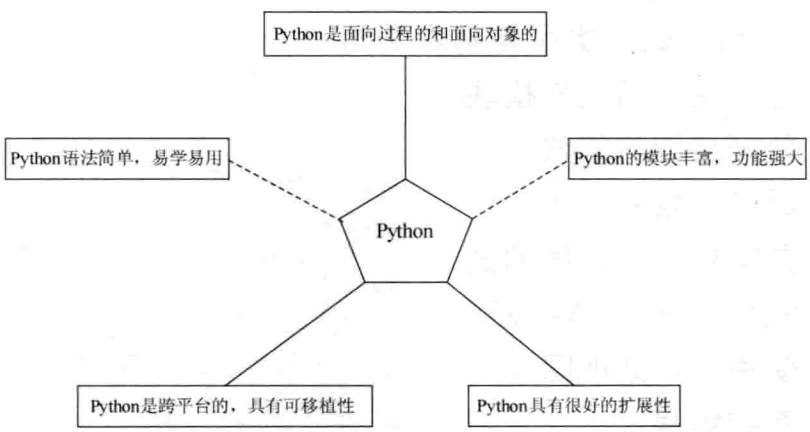


图 1-1　Python 的主要特点

Python 最大的特点是其独特而又简洁的语法。在 Python 源代码中，通过代码在行中不同的缩进量来表示代码所属的语句块，从而不需要使用类似 C/C++中的花括号“{}”。这个特点对于学过 C/C++，或长期使用 C/C++的用户来说，可能会有一点不适应。但是，Python 强制性的缩进可使代码看起来更加清晰，并且使用缩进在一定程度上也减少了代码输入量（因为省去了一对花括号），当然，前提是所使用的代码编辑器支持自动缩进。

使用 Python 可以非常快速地进行编程开发。Python 的语法很简单，对于初学者来说，学习并使用 Python 并不困难。甚至仅花费几天时间对 Python 进行短暂的学习，就可以使用 Python 编写出“引以为傲”的非常实用的脚本。

作为脚本语言，Python 非常灵活，使用 Python 可以实现各种各样的功能。通常来说，脚本语言都是解释型的语言，不需要编译过程。虽然 Python 被称之为脚本语言，但 Python 具有编译过程，Python 会将脚本编译成字节码的形式。并且，一般不需要程序员对脚本显式地进行编译，Python 自己会根据需要编译（通常情况下，只有作为模块的脚本才会被编译成字节码的形式）。

Python 遵循 GPL 协议，是源代码开放的软件。用户不仅可以免费使用 Python 来编写脚本，还可以阅读 Python 的源代码，了解 Python 的内部代码。用户甚至还可以参与到 Python 的开发之中，为 Python 的发展做出贡献。

1.2 Python 有什么优点

虽然 Python 不是程序语言唯一的选择，但却是一种不错的选择。为什么使用 Python 作为编程语言？这首先取决于用户。面对简单易学且功能强大的 Python，越来越多的人选择使用它来完成各种各样的任务。

1. Python 是免费的自由软件

Python 遵循 GPL 协议，是自由软件，这是 Python 流行的原因之一。用户使用 Python 进行开发和发布自己编写的程序不需要支付任何费用，不用担心版权问题。即使作为商业用途，Python 也是免费的。开源的自由软件正在成为软件行业的一种发展趋势，现在，很多商业软件公司也开始将自己的产品变为开源的，如 Java。作为开源软件的 Python 将具有更强的生命力。

作为自由软件，最令人鼓舞的就是可以很方便地获取源代码。程序员通过阅读其源代码，发现其中的神奇之处。

当深入地使用 Python 以后，可能会发现 Python 的某些特性，而这些特性并没有详细的说明文档，此时可以阅读 Python 的源代码去详细地了解 Python 的这些特性。

2. Python 是跨平台的

跨平台、良好的可移植性是 C 语言成为经典编程语言的关键，而 Python 正是用可移植的 ANSI C 编写的，这意味着 Python 也具有良好的跨平台特性。也就是说，在 Windows 下编写的 Python 脚本可以轻易地运行在 Linux 下。当然，如果在 Python 脚本中使用了 Windows 的某些特性（如 COM），那就另当别论了。

Python 不仅能在 Windows、Linux 系统中运行，由于它的开源本质，已经被移植在许多平台上，包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE、PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。不难看出，Python 不仅能够运行在传统的 Server、PC 系统中，还能够运行在正蓬勃发展的各类移动系统中。

3. Python 功能强大

Python 强大的功能也许才是很多用户支持 Python 的最重要的原因。从字符串处理到复杂的 3D 图形编程，Python 借助扩展模块都可以轻松完成。实际上，Python 的核心模块已经提供了足够强

大的功能，使用 Python 精心设计的内置对象可以完成许多功能强大的操作。

基于强大的功能，Python 可以使用在众多领域，例如：

- 系统编程，通过 Python 编程可帮助用户完成烦琐的日常工作；
- 科学计算，Python 简洁的语法可以像使用计算器一样来完成科学计算；
- 快速原型，Python 省去了编译调试的过程，可以快速地实现系统原型；
- Web 编程，使用 Python 可以编写 CGI，而现在流行的 Web 框架也可以使用 Python 实现。

4. Python 是可扩展的

Python 提供了扩展接口，通过使用 C/C++可以为 Python 编写扩展。另外，Python 还可以嵌入到 C/C++编写的程序之中。在 C/C++编写的程序之中可以使用 Python 完成一些对于 C/C++实现起来较为复杂的任务。在某些情况下，Python 可以作为动态链接库的替代品在 C/C++中使用。

Python 可以很容易地被修改、调试，而不需要重新编译。使用 Python 也不必担心版本的问题。

5. Python 易学易用

Python 的语法十分简单，而且在 Python 中数据类型的概念十分模糊。在使用变量时无须事先声明变量的类型。

使用 Python 不必关心内存的使用和管理，Python 会自动地分配、回收内存。

Python 提供了功能强大的内置对象和方法。使用 Python 可以减少其他编程语言的复杂性。例如，在 C 语言中使用数十行代码实现的排序。而在 Python 中，可以通过列表的排序函数轻松完成。通过几天的学习，加上对 Python 模块函数的了解，便可以使用 Python 很快地编写出功能强大的脚本。

1.3 其他程序设计语言中的 Python

由于 Python 代码具有简明、方便和易读等特性，因此，大部分高级程序设计语言也提供了对 Python 的实现支持，如在 Java 平台中实现了 Jython、在.NET 平台中实现了 IronPython 等。

1. Jython

Jython，旧称 Jpython，是 Python 语言的 Java 实现。

Java 是一种面向对象的程序设计语言。Java 基本上是仿照 C++进行开发的，与 C++不同的是，Java 是完全面向对象的，在 Java 中，舍弃了 C++中容易导致问题的指针。另外，Java 提供了自动垃圾回收的功能，用于回收不再被使用的内存。Java 和 Python 一样，都是解释性语言，并且 Java 也是跨平台的，而且 Java 和 Python 都能将代码编译。但相对于 Python 而言，Java 则过于庞大和复杂。

Jython 是由 Jim Hugunin 使用 Java 对“原始” Python 的重写。由于 Jython 的出现，为了避免混淆，将通常所说的 Python 也称之为 CPython(因为其是由 C 编写的)。CPython 是相对于 Jython 而言的。Jython 的出现使得 Python 可以在 Java 环境中运行。Jython 和 CPython 的运行方式相同，在 Jython 下编写 Python 脚本和在 CPython 下编写 Python 脚本并没有太多的区别。

相对于 CPython 而言，Jython 要慢一些。这是因为 Java 是解释型的语言，Jython 本身首先要由 Java 解释器进行解释，而编写的 Python 脚本又需要由 Jython 解释器进行解释，这当然要比 CPython 慢一些。多数情况下，这并不是什么大问题。由于 Java 的流行，有大量的 Java 程序可以

使用。有了 Jython，就可以在 Python 中非常方便地使用 Java 丰富的链接库。除此之外，使用 Jython 还可以在 Java 中使用 Python 快速、简便地进行开发，充分使用 Python 的灵活和快速。

2. Python for .NET

.NET 是 Microsoft XML Web Services 平台，是微软所极力推崇的技术。使用 XML Web Services，不同的应用程序可以进行通信和数据共享。即使这些应用程序运行在不同的操作系统上，也不会影响它们之间的通信和数据共享。

随着.NET 的流行，出现了 Python for .NET 和 IronPython 这两种.NET 平台上的 Python。

Python for .NET 可以使用 Python 与.NET 进行交互，在.NET 中使用 Python 作为脚本语言，而且通过 Python for .NET，Python 也可以使用.NET 中的服务和组件。

3. IronPython

IronPython 也是 Python 编程语言在.NET 平台上的实现，由 Jim Hugunin（同时也是 Jython 创造者）所创造。IronPython 提供了和 Python 一样的交互式命令行。在交互式命令行下，可以使用 Python 访问所有的.NET 库。IronPython 除了对 Python 语言完全兼容外，还支持所有的.NET 类型库，并且继承了.NET Framework 的优点。通过 IronPython 可以使用 Python 来扩展.NET。

与 Python for .NET 相比，IronPython 的功能更加强大。IronPython 支持静态编译，通过静态编译，可以将 IronPython 程序编译成常规的.NET 的可执行文件。另外还可以将 IronPython 程序静态编译为.NET 动态链接库，所编译的动态链接库可以被 C#、VB.NET 等调用。

不管是 Python for .NET 还是 IronPython，都还不完善。但是，借助于开源的神奇力量，相信 Python 在.NET 上完善地使用只是时间上的问题。而且，微软的.NET 在开源社区也很受欢迎，甚至在 Linux 上也可以运行.NET 程序。

1.4 快速搭建 Python 开发环境

Python 可以运行在多种操作系统上，本书是以 Windows 为平台来介绍 Python。Python 官方网站提供了 Windows 下的安装程序，通过运行安装程序可以非常方便地安装 Python。除了安装 Python 以外，还应该选择用于编辑 Python 脚本的文本编辑器，提高编辑脚本的效率。因为在 Python 中，是使用缩进来表示语句块，因此所选择的文本编辑器最后具有自动缩进功能。

1.4.1 哪些系统中可使用 Python

本章前面已介绍过 Python 是跨平台的，可以运行在绝大多数的操作系统中。程序员平常用得比较多的操作系统主要包括以下几种：

- Windows
- Linux/UNIX
- Mac OS X
- OS/2

这些操作系统都可以使用 Python，因此，对于普通的用户而言，一般不需要担心自己所使用的系统不能运行 Python（绝大多数的流行操作系统都可以使用 Python）。本书以 Windows 7 操作

系统为平台，主要使用 Python 3.2 进行讲解（Python 现在最新版已是 3.4 了，但由于很多扩展库都还不支持最新版，因此，本书选用 Python 3.2 进行介绍）。

1.4.2 Python 的下载和安装

Python 的安装程序以及源代码都可以从其官方网站 http://www.python.org/获取。下面以 Windows 7、Python 3.2 为例，介绍在 Windows 下安装 Python 的过程，具体操作步骤如下。

step 1 从 Python 官方网站 http://www.python.org/下载 Python 在 Windows 下的安装程序 python-3.2.5.msi。

step 2 双击下载的安装程序进行安装，显示如图 1-2 所示界面。

step 3 如果系统中存在多个用户，而其他用户并不需要使用 Python，则可以选中【Install just for me】单选按钮，否则按照默认选项设置（所有用户都可使用 Python）。

step 4 单击【Next】按钮，显示如图 1-3 所示界面，设置安装路径。此处既可以按照默认安装路径，也可以根据需要选择 Python 的安装路径。

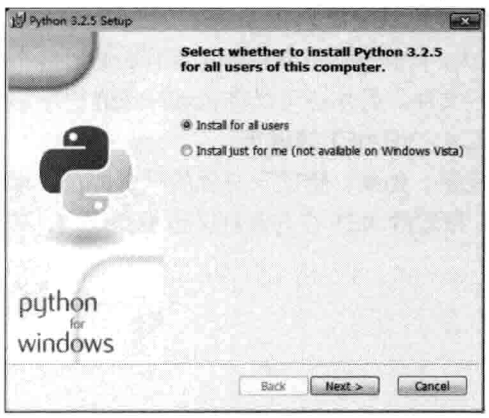


图 1-2 Python 安装程序

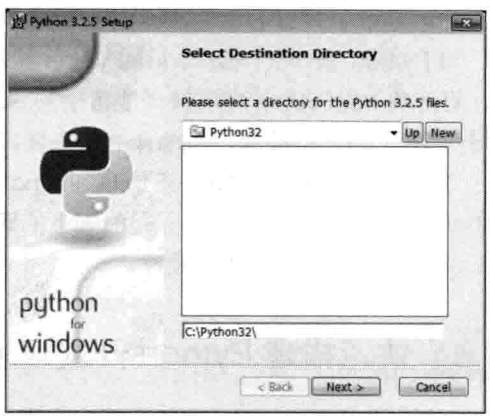


图 1-3 选择安装路径

step 5 单击【Next】按钮，进入选择安装模块界面，如图 1-4 所示。此处不需要修改，按照默认配置即可。

step 6 单击【Next】按钮，Python 安装程序将开始复制安装文件，如图 1-5 所示。

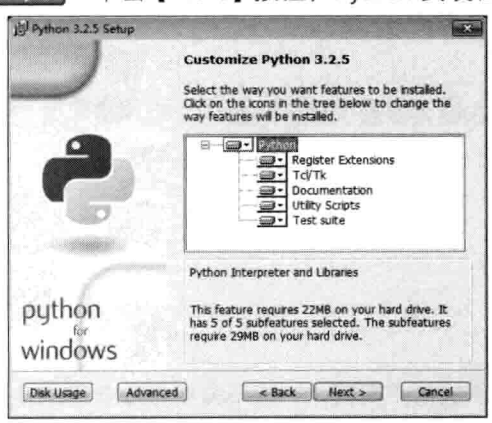


图 1-4 选择安装模块

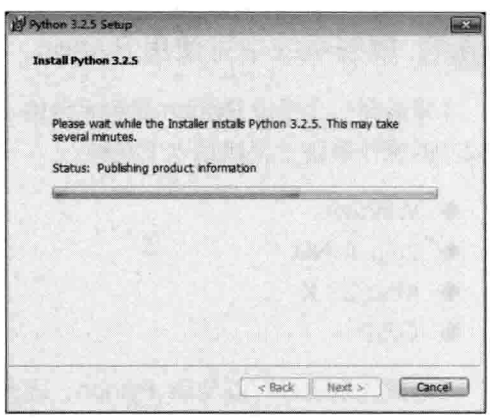


图 1-5 复制安装文件