

Beginning Python From Novice to Professional **Second Edition**

Python基础教程

(第2版·修订版)

[挪] Magnus Lie Hetland ◎著
司维 曾军崑 谭颖华 ◎译
钟读杭 ◎审校

给真正的初学者写的入门书

全面详尽，10个项目引人入胜

TURING

图灵程序设计丛书

014043509

TP311.56

1030-2

Beginning Python From Novice to Professional **Second Edition**

Python基础教程

(第2版·修订版)

[挪] Magnus Lie Hetland ◎著

司维 曾军崑 谭颖华 ◎译

钟读杭 ◎审校



北航

C1731819

人民邮电出版社

北京

TP311.56

1030-2

图书在版编目 (CIP) 数据

Python基础教程 / (挪) 海特兰德 (Hetland, M. L.)
著; 司维, 曾军崑, 谭颖华译. -- 2版 (修订本). --
北京: 人民邮电出版社, 2014. 6
(图灵程序设计丛书)
ISBN 978-7-115-35352-8

I. ①P… II. ①海… ②司… ③曾… ④谭… III. ①
软件工具—程序设计—教材 IV. ①TP311.56

中国版本图书馆CIP数据核字(2014)第072685号

内 容 提 要

本书包括 Python 程序设计的方方面面, 首先从 Python 的安装开始, 随后介绍了 Python 的基础知识和基本概念, 包括列表、元组、字符串、字典以及各种语句。然后循序渐进地介绍了一些相对高级的主题, 包括抽象、异常、魔法方法、属性、迭代器。此后探讨了如何将 Python 与数据库、网络、C 语言等工具结合使用, 从而发挥出 Python 的强大功能, 同时介绍了 Python 程序测试、打包、发布等知识。最后, 作者结合前面讲述的内容, 按照实际项目开发的步骤向读者介绍了几个具有实际意义的 Python 项目的开发过程。

本书内容涉及的范围较广, 既能为初学者夯实基础, 又能帮助程序员提升技能, 适合各个层次的 Python 开发人员阅读参考。

-
- ◆ 著 [挪] Magnus Lie Hetland
 - 译 司 维 曾军崑 谭颖华
 - 审 校 钟读杭
 - 责任编辑 朱 巍
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 30.5
 - 字数: 727千字 2014年6月第2版
 - 印数: 39 501 - 44 500册 2014年6月北京第1次印刷
 - 著作权合同登记号 图字: 01-2009-2890号
-

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版 权 声 明

Original English language edition, entitled *Beginning Python: From Novice to Professional, Second Edition* by Magnus Lie Hetland, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Magnus Lie Hetland. Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L. P. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

前 言

新版的《Python基础教程》终于和大家见面了。如果算上这本书的前身*Practical Python*，实际上这已经是第3版了。这本书也让我投入了近10年时间。这期间，Python发生了许多有趣的变化，我也在尽力更新我对这门语言的介绍。同时，Python也面临着在相当长的一段时间内最具标志性的变化：推出了第3版。在本书撰写时，最终版本还没有发布，但是其特性已跃然纸上，并且已经推出了几个可用的版本。这次升级带来一个很有意思的挑战，就是它不再向下兼容。换句话说，它并不是简单地增加一些功能，让我可以挑挑捡捡地写进书中。它同时还会改变已经存在的语言特性，也就是说Python 2.5版本中大家习以为常的事情在新版本将不再成立。

要是整个Python社区都能立即着手转换到新版本，更新所有遗留代码，那当然不会有什么问题。我只要讲解新的语言就可以了！但是，大量用旧版编写的代码还会存在，并且人们仍然有可能继续用旧版编写代码，直到第3版完全板上钉钉。

那么，怎么应付这个变化呢？首先，就算新版本有些不兼容的改变，但语言的大部分还是相同的。也就是说，如果我的程序完全用Python 2.5编写，基本上在Python 3上也能运行（在兼容版本2.6版上更不成问题）。至于那些不再正确的部分，我则持保守态度，猜测Python 3完全被大家接受还需要一定时间。本书基本上基于2.5版本编写，并且在行文中标注出在新版中会有改变的部分。除此之外，我还增加了一个附录D，概要介绍了一些主要改变。对于大多数读者来说，我觉得应该够了。

在撰写本书第2版时，我得到了许多人的帮助。就像在写前两版（本书第1版和之前的*Practical Python*）一样，Jason Gilmore扶我上马，并且在项目进展的过程中扮演了重要角色。Richard Dal Porto、Frank Pohlmann和Dominic Shakeshaft在过程中也常助我一臂之力。Richard Taylor至关重要，他要确保代码完全正确（如果还是有错，大家骂我好了），而Marilyn Smith为我的写作润色不少。我还要感谢Apress的其他同仁们，包括Liz Berry、Beth Christmas、Steve Anglin和Tina Nielsen，还有那些帮我修订错误以及提出宝贵意见的读者们，其中包括Bob Helmbold和Waclaw Kusnierczyk。当然，我还要感谢所有那些将本书的前两版本买回家的读者们。

第1版前言

几年前，Jason Gilmore建议我为Apress出版社写本书。他读了我的在线Python教程，希望我以类似的风格撰写一本书。我受宠若惊，也很兴奋，还有点紧张。最让我担心的就是写书可能要花费很长时间，以及它对于我的学业（我当时是博士生）会有影响。结果这件事成了一项艰巨的任务，而且花的时间比我预期的长很多。幸运的是，它没有过多地影响我的学习，我也按时获得了博士学位。

去年，Jason又联系了我。Apress出版社希望修订本书，问我是否有兴趣。那时候我正忙着熟悉新的副教授职位，而几乎所有的业余时间都用来扮演皮尔·金特（Peer Gynt）了，所以时间又变成了主要问题。最终，在事情安排妥当一些，而且我也有更多的业余时间之后，我同意做修订，而本书（我相信你已经猜到了）就是最终的成果。大多数资料都是从本书的第1版*Practical Python*中拿来的。本书基于Python语言的最新更新，对现有的内容进行了全面的修订，增加了几个新章节。有些旧的内容也进行了重新分配，以适应新的结构。我从读者那里得到了不少关于第1版的正面反馈，所以我希望能够继续保留读者所喜爱的特点，并且锦上添花。

如果没有其他人对我的持续帮助和鼓励，这本书是无法完成的。我衷心地感谢他们。特别要感谢在本书撰写过程中和我一起工作的团队：Jason Gilmore，感谢他落实了这个项目并且将项目引导至正确的方向；Beckie Stones，感谢她将所有内容整理在一起；Jeremy Jones和Matt Moodie，感谢他们专业的意见和洞察力；Linda Marousek，感谢她对我如此耐心。我还要感谢团队中的其他人，感谢他们让这个过程变得如此顺利。但是如果没有那些在前一版本中和我一起工作的人所付出的努力，这本书也是无法完成的。我要感谢Jason Gilmore和Alex Martelli，感谢他们杰出的技术编辑工作（Jason负责整本书，Alex负责前半部分）以及工作职责之外的修改意见和建议；Erin Mulligan和Tory McLearn，感谢他们在撰写过程中和我共同进退，在我需要的时候给予我敦促；Nancy Rapoport，感谢她对我的文稿进行润色；Grace Wong，感谢她在别人无法回答的时候给予我答案。Pete Shinnars对项目10的游戏给过我一些有帮助的建议，我非常感谢她。对本书感到满意的读者也给我发来了邮件，这极大地鼓舞了我的斗志，感谢你们！最后，我要谢谢我的家人和朋友，以及我的女朋友Ranveig。在撰写本书的过程中，他们都一直宽容地陪伴着我。

开 场 白

编写C程序就像一群人拿着剃刀在刚打过蜡的舞场内跳快舞。

——Waldi Ravens

C++：难学更难用，设计如此。

——匿名

在很多方面，Java就是C++。

——Michael Feldman

现在请看一种前所未有的表演……

——Monty Python的表演《飞行的马戏团》

我引用了别人的几句话作为本书的开篇，目的是为本书奠定一个很不正式的基调。为了让大家可以轻松地阅读，我试图用一点儿幽默的方式来讨论Python编程的主题。幽默是Python社区的传统，而很多幽默都和Monty Python^①的滑稽短剧有关。所以我举的一些例子可能看起来有些傻，希望你能够接受。（顺便说一句，Python这个名字是从Monty Python来的，而不是源于蟒蛇这种动物。）

在这个开场白中，我会简单地告诉你Python是什么，为什么你应该使用它，谁在使用它，本书写给谁看以及本书的行文结构。

那么，什么是Python？为什么你应该使用它呢？还是引用官方的说法吧（<http://www.python.org/doc/essays/blurb.html>），Python就是“一种解释型的、面向对象的、带有动态语义的高级程序设计语言”。这句话中的很多术语，你可以在阅读本书的过程中逐渐弄懂，但最重要的是，Python是一种让你在编程实现自己想法时感觉不那么碍手碍脚的程序设计语言。你可以花较少的代价实现想要的功能，并且编写的程序清晰易懂（和当前流行的其他各种程序设计语言相比更是如此）。

尽管Python运行起来可能不会像C或者C++那样的编译型语言一样快，但是Python让你节省下来的编程时间足以成为它值得一用的理由，何况大多数程序的运行速度差异可能并不是那么显而易见。如果你是C程序员，你可以用C实现制约程序性能的关键部分，并且轻松地让它们和使用

^① Monty Python是20世纪70年代风靡全球的英国六人喜剧团体。——译者注

Python编写的部分相互协作。如果你之前没有任何编程的经历（而且可能被我前面提到的C和C++的话唬住了），那么既简单又强大的Python就是你入门的完美选择。

那么谁在使用Python呢？自从Guido van Rossum在20世纪90年代初创造这门语言以来，它的追随者就一直在稳步增加，而且近些年来，社区对它的兴趣也日益浓厚。Python广泛用于系统管理工作（比如它是很多Linux发行版的重要组成部分），也可以作为从零起步的入门编程语言。NASA在它的几个系统中既用Python做主程序开发，又将其作为脚本语言。Industrial Light & Magic（工业光魔公司）在高预算影片中使用Python制作影片的特效；Yahoo!使用它（包括其他技术）管理讨论组；Google用它实现网络爬虫和搜索引擎中的很多组件。Python也被用于计算机游戏和生物信息等各种领域。不久后可能就会有人问了：“谁不用Python呢？”

本书面向那些希望学习如何使用Python来编程的读者，其内容适合广泛的读者群，不管你是程序设计的学徒还是高级的计算机魔法师。如果你之前从未写过程序，那么你应该从第1章开始阅读，直到觉得所学的内容对你来说有些超前了（我是说如果有可能这样），此时你就应该开始实践，编写自己的程序，待时机成熟，你就可以再回到书本上，学习那些更复杂的内容。

如果你已经知道如何编程，那么，你对一般介绍性的内容可能不会感到新鲜了（但是，文中可能到处都有一些让你惊讶的细节）。你可以快速阅读前面的章节，从而了解Python是如何工作的，或者阅读附录A，它是基于我的在线Python教程Instant Python写成的。它会让你快速了解很多重要的Python概念。有了大概印象后，你可以直接跳到第10章（讨论Python标准库）。

本书的最后几章包括10个程序设计项目，从多方面展示了Python语言的能力。不管是新手还是专家，都会对这些项目感兴趣。尽管后面项目中的一些内容对于没有经验的编程者来说可能有些难，但是（在读完本书的前半部分后）按照项目顺序完成应该还是可能的。

这些项目包括了多方面的主题，多数对你编写自己的程序都很有用。你可以学会做一些现在可能完全不会做的事情，比如创建聊天服务器、点对点的文件共享系统或者功能完备的计算机图形游戏等。乍一看，你会觉得很多内容都很难，但是到了最后，我想你会惊讶地发现它们实际上是如此简单。如果你想下载源代码，可以访问<http://www.apress.com>。

好了，开场白到此为止，我个人觉得冗长的介绍比较无聊，让我们马上开始学习Python编程吧，从第1章或者附录A开始。祝你好运，改造愉快。

目 录

第 1 章 快速改造：基础知识	1	1.11.5 长字符串、原始字符串和 Unicode	22
1.1 安装 Python	1	1.12 小结	24
1.1.1 Windows	1	1.12.1 本章的新函数	25
1.1.2 Linux 和 UNIX	3	1.12.2 接下来学什么	25
1.1.3 苹果机 (Macintosh)	4	第 2 章 列表和元组	26
1.1.4 其他发行版本	5	2.1 序列概览	26
1.1.5 时常关注，保持更新	6	2.2 通用序列操作	27
1.2 交互式解释器	6	2.2.1 索引	27
1.3 算法是什么	7	2.2.2 分片	29
1.4 数字和表达式	8	2.2.3 序列相加	31
1.4.1 长整数	9	2.2.4 乘法	31
1.4.2 十六进制和八进制	10	2.2.5 成员资格	32
1.5 变量	10	2.2.6 长度、最小值和最大值	33
1.6 语句	11	2.3 列表：Python 的“苦力”	34
1.7 获取用户输入	12	2.3.1 list 函数	34
1.8 函数	13	2.3.2 基本的列表操作	34
1.9 模块	14	2.3.3 列表方法	36
1.9.1 cmath 和复数	14	2.4 元组：不可变序列	41
1.9.2 回到 <code>__future__</code>	15	2.4.1 tuple 函数	42
1.10 保存并执行程序	15	2.4.2 基本元组操作	42
1.10.1 通过命令提示符运行 Python 脚本	16	2.4.3 那么，意义何在	42
1.10.2 让脚本像普通程序一样运行	17	2.5 小结	43
1.10.3 注释	18	2.5.1 本章的新函数	43
1.11 字符串	19	2.5.2 接下来学什么	43
1.11.1 单引号字符串和转义引号	19	第 3 章 使用字符串	44
1.11.2 拼接字符串	20	3.1 基本字符串操作	44
1.11.3 字符串表示，str 和 repr	20	3.2 字符串格式化：精简版	44
1.11.4 input 和 raw_input 的比较	21	3.3 字符串格式化：完整版	46
		3.3.1 简单转换	47

3.3.2	字段宽度和精度	47	5.5	循环	76
3.3.3	符号、对齐和用 0 填充	47	5.5.1	while 循环	77
3.4	字符串方法	49	5.5.2	for 循环	77
3.4.1	find	49	5.5.3	循环遍历字典元素	78
3.4.2	join	50	5.5.4	一些迭代工具	79
3.4.3	lower	51	5.5.5	跳出循环	80
3.4.4	replace	51	5.5.6	循环中的 else 子句	82
3.4.5	split	52	5.6	列表推导式——轻量级循环	82
3.4.6	strip	52	5.7	三人行	83
3.4.7	translate	52	5.7.1	什么都没发生	84
3.5	小结	54	5.7.2	使用 del 删除	84
3.5.1	本章的新函数	54	5.7.3	使用 exec 和 eval 执行和求值 字符串	85
3.5.2	接下来学什么	54	5.8	小结	87
第 4 章	字典：当索引不好用时	55	5.8.1	本章的新函数	88
4.1	字典的使用	55	5.8.2	接下来学什么	88
4.2	创建和使用字典	56	第 6 章	抽象	89
4.2.1	dict 函数	56	6.1	懒惰即美德	89
4.2.2	基本字典操作	57	6.2	抽象和结构	90
4.2.3	字典的格式化字符串	58	6.3	创建函数	90
4.2.4	字典方法	59	6.3.1	文档化函数	91
4.3	小结	64	6.3.2	并非真正函数的函数	92
4.3.1	本章的新函数	64	6.4	参数魔法	92
4.3.2	接下来学什么	64	6.4.1	值从哪里来	93
第 5 章	条件、循环和其他语句	65	6.4.2	我能改变参数吗	93
5.1	print 和 import 的更多信息	65	6.4.3	关键字参数和默认值	97
5.1.1	使用逗号输出	65	6.4.4	收集参数	99
5.1.2	把某件事作为另一件事导入	66	6.4.5	参数收集的逆过程	101
5.2	赋值魔法	67	6.4.6	练习使用参数	102
5.2.1	序列解包	67	6.5	作用域	103
5.2.2	链式赋值	68	6.6	递归	105
5.2.3	增量赋值	68	6.6.1	两个经典：阶乘和幂	106
5.3	语句块：缩排的乐趣	69	6.6.2	另外一个经典：二分法查找	107
5.4	条件和条件语句	69	6.7	小结	110
5.4.1	这就是布尔变量的作用	69	6.7.1	本章的新函数	110
5.4.2	条件执行和 if 语句	70	6.7.2	接下来学什么	111
5.4.3	else 子句	71	第 7 章	更加抽象	112
5.4.4	elif 子句	71	7.1	对象的魔力	112
5.4.5	嵌套代码块	71	7.1.1	多态	113
5.4.6	更复杂的条件	72	7.1.2	封装	115
5.4.7	断言	76			

7.1.3 继承	117	9.3.2 子类化列表, 字典和字符串	146
7.2 类和类型	117	9.4 更多魔力	147
7.2.1 类到底是什么	117	9.5 属性	147
7.2.2 创建自己的类	118	9.5.1 property 函数	148
7.2.3 特性、函数和方法	119	9.5.2 静态方法和类成员方法	149
7.2.4 类的命名空间	121	9.5.3 <code>__getattr__</code> 、 <code>__setattr__</code> 和 它的朋友们	150
7.2.5 指定超类	122	9.6 迭代器	151
7.2.6 检查继承	122	9.6.1 迭代器规则	151
7.2.7 多个超类	123	9.6.2 从迭代器得到序列	153
7.2.8 接口和内省	124	9.7 生成器	153
7.3 一些关于面向对象设计的思考	125	9.7.1 创建生成器	153
7.4 小结	125	9.7.2 递归生成器	154
7.4.1 本章的新函数	126	9.7.3 通用生成器	156
7.4.2 接下来学什么	126	9.7.4 生成器方法	156
9.7.5 模拟生成器	157	9.7.5 模拟生成器	157
第 8 章 异常	127	9.8 八皇后问题	158
8.1 什么是异常	127	9.8.1 生成器和回溯	158
8.2 按自己的方式出错	127	9.8.2 问题	159
8.2.1 <code>raise</code> 语句	127	9.8.3 状态表示	159
8.2.2 自定义异常类	128	9.8.4 寻找冲突	159
8.3 捕捉异常	129	9.8.5 基本情况	160
8.4 不止一个 <code>except</code> 子句	130	9.8.6 需要递归的情况	161
8.5 用一个块捕捉两个异常	131	9.8.7 打包	162
8.6 捕捉对象	131	9.9 小结	163
8.7 真正的全捕捉	132	9.9.1 本章的新函数	164
8.8 万事大吉	132	9.9.2 接下来学什么	164
8.9 最后	134	第 10 章 自带电池	165
8.10 异常和函数	134	10.1 模块	165
8.11 异常之禅	135	10.1.1 模块是程序	165
8.12 小结	136	10.1.2 模块用于定义	167
8.12.1 本章的新函数	137	10.1.3 让你的模块可用	169
8.12.2 接下来学什么	137	10.1.4 包	171
第 9 章 魔法方法、属性和迭代器	138	10.2 探究模块	172
9.1 准备工作	138	10.2.1 模块中有什么	172
9.2 构造方法	139	10.2.2 用 <code>help</code> 获取帮助	173
9.2.1 重写一般方法和特殊的构造 方法	140	10.2.3 文档	174
9.2.2 调用未绑定的超类构造方法	141	10.2.4 使用源代码	174
9.2.3 使用 <code>super</code> 函数	142	10.3 标准库: 一些最爱	175
9.3 成员访问	143	10.3.1 <code>sys</code>	175
9.3.1 基本的序列和映射规则	144	10.3.2 <code>os</code>	176

10.3.3	fileinput	178	12.4	但是我宁愿用	228
10.3.4	集合、堆和双端队列	180	12.4.1	使用 Tkinter	229
10.3.5	time	184	12.4.2	使用 Jython 和 Swing	229
10.3.6	random	185	12.4.3	使用其他开发包	230
10.3.7	shelve	188	12.5	小结	230
10.3.8	re	191	第 13 章 数据库支持		231
10.3.9	其他有趣的标准模块	204	13.1	Python 数据库编程接口 (API)	232
10.4	小结	205	13.1.1	全局变量	232
10.4.1	本章的新函数	205	13.1.2	异常	233
10.4.2	接下来学什么	206	13.1.3	连接和游标	233
第 11 章 文件和流		207	13.1.4	类型	235
11.1	打开文件	207	13.2	SQLite 和 PySQLite	235
11.1.1	文件模式	207	13.2.1	入门	236
11.1.2	缓冲	208	13.2.2	数据库应用程序示例	237
11.2	基本的文件方法	209	13.3	小结	239
11.2.1	读和写	209	13.3.1	本章的新函数	240
11.2.2	管式输出	210	13.3.2	接下来学什么	240
11.2.3	读写行	211	第 14 章 网络编程		241
11.2.4	关闭文件	212	14.1	少数几个网络设计模块	241
11.2.5	使用基本文件方法	213	14.1.1	socket 模块	242
11.3	对文件内容进行迭代	214	14.1.2	urllib 和 urllib2 模块	243
11.3.1	按字节处理	214	14.1.3	其他模块	245
11.3.2	按行操作	215	14.2	SocketServer 和它的朋友们	246
11.3.3	读取所有内容	215	14.3	多个连接	247
11.3.4	使用 fileinput 实现懒惰 行迭代	216	14.3.1	使用 SocketServer 进行 分叉和线程处理	248
11.3.5	文件迭代器	216	14.3.2	带有 select 和 poll 的 异步 I/O	249
11.4	小结	217	14.4	Twisted	251
11.4.1	本章的新函数	218	14.4.1	下载并安装 Twisted	251
11.4.2	接下来学什么	218	14.4.2	编写 Twisted 服务器	251
第 12 章 图形用户界面		219	14.5	小结	253
12.1	丰富的平台	219	14.5.1	本章的新函数	254
12.2	下载和安装 wxPython	220	14.5.2	接下来学什么	254
12.3	创建示例 GUI 应用程序	221	第 15 章 Python 和 Web		255
12.3.1	开始	221	15.1	屏幕抓取	255
12.3.2	窗口和组件	222	15.1.1	Tidy 和 XHTML 解析	256
12.3.3	标签、标题和位置	223	15.1.2	Beautiful Soup	260
12.3.4	更智能的布局	225	15.2	使用 CGI 创建动态网页	261
12.3.5	事件处理	226			
12.3.6	完成了的程序	226			

15.2.1	第一步：准备网络服务器	261	第 17 章 扩展 Python	291
15.2.2	第二步：加入 Pound Bang 行	261	17.1 考虑哪个更重要	291
15.2.3	第三步：设置文件权限	262	17.2 非常简单的途径：Jython 和 IronPython	292
15.2.4	CGI 安全风险	263	17.3 编写 C 语言扩展	294
15.2.5	简单的 CGI 脚本	263	17.3.1 SWIG	296
15.2.6	使用 cgitb 调试	263	17.3.2 自己研究	299
15.2.7	使用 cgi 模块	264	17.4 小结	303
15.2.8	简单的表单	266	17.4.1 本章的新函数	304
15.3	更进一步：mod_python	267	17.4.2 接下来学什么	304
15.3.1	安装 mod_python	268	第 18 章 程序打包	305
15.3.2	CGI 处理程序	269	18.1 Distutils 基础	305
15.3.3	PSP	270	18.2 打包	307
15.3.4	发布	271	18.2.1 建立存档文件	307
15.4	网络应用程序框架	273	18.2.2 创建 Windows 安装程序或 RPM 包	308
15.5	Web 服务：正确分析	274	18.3 编译扩展	309
15.5.1	RSS 和相关内容	274	18.4 使用 py2exe 创建可执行程序	310
15.5.2	使用 XML-RPC 进行远程 过程调用	275	18.5 小结	311
15.5.3	SOAP	276	18.5.1 本章的新函数	311
15.6	小结	276	18.5.2 接下来学什么	311
15.6.1	本章的新函数	276	第 19 章 好玩的编程	312
15.6.2	接下来学什么	276	19.1 为什么要好玩	312
第 16 章 测试	277		19.2 程序设计的柔术	312
16.1	先测试，后编码	277	19.3 原型设计	313
16.1.1	精确的需求说明	278	19.4 配置	314
16.1.2	为改变而计划	279	19.4.1 提取常量	314
16.1.3	测试的 4 个步骤	279	19.4.2 配置文件	315
16.2	测试工具	280	19.5 日志记录	317
16.2.1	doctest	280	19.6 如果还没烦	318
16.2.2	unittest	282	19.7 如果还想学	318
16.3	单元测试以外的内容	285	19.8 小结	319
16.3.1	使用 PyChecker 和 PyLint 检查源代码	285	第 20 章 项目 1：即时标记	320
16.3.2	性能分析	288	20.1 问题是什么	320
16.4	小结	289	20.2 有用的工具	321
16.4.1	本章的新函数	290	20.3 准备工作	321
16.4.2	接下来学什么	290	20.4 初次实现	322

20.4.1	找出文本块	322	23.2	有用的工具	360
20.4.2	添加一些标记	323	23.3	准备工作	360
20.5	再次实现	324	23.4	初次实现	360
20.5.1	处理程序	325	23.5	再次实现	363
20.5.2	处理程序的超类	325	23.6	进一步探索	369
20.5.3	规则	327	第 24 章 项目 5: 虚拟茶话会		371
20.5.4	规则的超类	328	24.1	问题	371
20.5.5	过滤器	328	24.2	有用的工具	371
20.5.6	分析器	328	24.3	准备工作	372
20.5.7	构造规则和过滤器	329	24.4	初次实现	373
20.5.8	整合	332	24.4.1	ChatServer 类	373
20.6	进一步探索	336	24.4.2	ChatSession 类	374
第 21 章 项目 2: 画幅好画		337	24.4.3	整合	376
21.1	问题	337	24.5	再次实现	377
21.2	有用的工具	338	24.5.1	基础命令解释	378
21.3	准备工作	338	24.5.2	房间	378
21.4	初次实现	339	24.5.3	登录和退出房间	379
21.4.1	用 ReportLab 画图	339	24.5.4	主聊天室	379
21.4.2	生成折线	340	24.5.5	新的服务器	380
21.4.3	编写原型	341	24.6	进一步探索	384
21.5	再次实现	342	第 25 章 项目 6: 使用 CGI 进行远程编辑		385
21.5.1	获取数据	342	25.1	问题	385
21.5.2	使用 LinePlot 类	343	25.2	有用的工具	385
21.6	进一步探索	344	25.3	准备工作	386
第 22 章 项目 3: 万能的 XML		345	25.4	初次实现	386
22.1	问题	345	25.5	再次实现	387
22.2	有用的工具	346	25.5.1	创建文件名表单	387
22.3	准备工作	347	25.5.2	创建编辑器的脚本	388
22.4	初次实现	348	25.5.3	编写保存脚本	389
22.4.1	创建简单的内容处理器	348	25.5.4	运行编辑器	390
22.4.2	生成 HTML 页面	350	25.6	进一步探索	391
22.5	再次实现	352	第 26 章 项目 7: 自定义电子公告板		393
22.5.1	调度程序的混入类	352	26.1	问题	393
22.5.2	实现首部、页脚和默认的处理程序	354	26.2	有用的工具	394
22.5.3	对目录的支持	354	26.3	准备工作	394
22.5.4	事件处理程序	355	26.4	初次实现	395
22.6	进一步探索	358	26.5	再次实现	398
第 23 章 项目 4: 新闻聚合		359	26.5.1	编写 Main 脚本	399
23.1	问题	359	26.5.2	编写 View 脚本	400

26.5.3 编写 Edit 脚本.....	401	28.4 初次实现.....	422
26.5.4 编写 Save 脚本.....	403	28.5 再次实现.....	424
26.5.5 尝试使用.....	404	28.6 进一步探索.....	427
26.6 进一步探索.....	404	第 29 章 项目 10: DIY 街机游戏.....	428
第 27 章 项目 8: 使用 XML-RPC		29.1 问题.....	428
进行文件共享.....	406	29.2 有用的工具.....	429
27.1 问题.....	406	29.2.1 pygame.....	429
27.2 有用的工具.....	407	29.2.2 pygame.locals.....	429
27.3 准备工作.....	407	29.2.3 pygame.display.....	430
27.4 初次实现.....	408	29.2.4 pygame.font.....	430
27.4.1 实现一个简单的节点.....	408	29.2.5 pygame.sprite.....	430
27.4.2 尝试使用首次实现.....	413	29.2.6 pygame.mouse.....	431
27.5 再次实现.....	414	29.2.7 pygame.event.....	431
27.5.1 创建客户端界面.....	414	29.2.8 pygame.image.....	431
27.5.2 引发异常.....	415	29.3 准备工作.....	431
27.5.3 验证文件名.....	416	29.4 初次实现.....	432
27.5.4 尝试使用再次实现.....	419	29.5 再次实现.....	434
27.6 进一步探索.....	420	29.6 进一步探索.....	442
第 28 章 项目 9: 文件共享 2——		附录 A 简明版本.....	443
GUI 版本.....	421	附录 B Python 参考手册.....	450
28.1 问题.....	421	附录 C 在线资源.....	461
28.2 有用的工具.....	421	附录 D Python 3.0.....	464
28.3 准备工作.....	421		

快速改造：基础知识

现在正是做些改造的好时候^①。通过阅读本章，你将学到如何控制计算机，方法是借助一门计算机能够理解的语言——Python。这里没什么特别难以理解的内容，只要对计算机基本的工作原理有些了解，就能跟着例子自己试着去做。本书会从极其简单的内容开始，把基础知识过一遍。不过Python是一门非常强大的语言，所以读者很快就可以完成一些相当高级的任务了。

本章首先会介绍如何安装所需的软件，然后讲一点点算法及其主要的组成。在这部分内容中，会有许多简短的例子（大多数只使用了简单的算术），读者可以自己在Python的交互式解释器（参见1.2节）中一试身手。读者从中可以学习到变量（variable）、函数（function）和模块（module）的知识。介绍完这些主题后，我将讲解如何编写和运行规模更大一些的程序。最后，本章将讲述字符串（string），它在几乎所有Python程序中都很重要。

1.1 安装 Python

在开始编程前，需要安装一些新软件。下面简要介绍如何下载和安装Python。如果想直接跳到安装过程的介绍而不看详细的向导，可以直接访问网址<http://www.python.org/download>，下载并安装Python的最新版本。

1.1.1 Windows

要在Windows系统中安装Python，请参照下面的步骤进行。

- (1) 打开Web浏览器，访问<http://www.python.org>。
- (2) 点击Download链接。
- (3) 读者在这里可以看到几条链接，名字类似于Python 2.5.x或Python 2.5.x Windows installer。点击Windows installer链接并且下载安装文件（如果在Itanium或者AMD平台上运行Python，就需

^① “改造”（Hacking）并不同于“破坏”（Cracking），后者描述的是计算机犯罪行为。两者经常被混为一谈。改造意味着在编程时获得乐趣。更多的信息，请看Eric Raymond的文章How to Become a Hacker（如何成为一名“改造者”），网址为<http://www.catb.org/~esr/faqs/hacker-howto.html>。

要选择相应的安装文件)。

注意 如果无法找到第3步提及的链接,那么就点击版本号最高的链接,名字类似于Python 2.5.x。对于Python 2.5,可以直接访问<http://www.python.org/2.5>,然后遵照针对Windows用户的提示。用户需要下载一个名为python-2.5.x.msi(或者名字类似的)的文件,其中2.5.x指最新的版本号。

(4) 将Windows Installer文件放在计算机的任何位置均可,比如C:\download\python-2.5.x.msi。(只要建立一个之后能找到的文件夹就行。)

(5) 在Windows资源管理器中双击运行所下载的文件,开启Python的安装向导。该向导使用起来很简单,只需要接受默认设置,等到安装程序完成,你就能使用Python了!

假如安装一切正常,应该能在Windows的开始菜单中找到新安装的Python程序。按照开始→程序→Python^①→IDLE(Python GUI)的步骤运行Python集成开发环境(Python Integrated Development Environment, IDLE)。

你现在应该看到一个类似于图1-1的窗口。如果感觉有点困惑,那么在菜单中选择Help→IDLE Help,会看到一个简单的说明,介绍各个菜单项及其基本用法。更多关于IDLE的文档的信息,请访问<http://www.python.org/idle>。(在这里也能看到如何在Windows之外的其他平台上运行IDLE的信息。)如果按下F1,或者选择Help→Python Docs,会启动完整的Python文档。对读者最有用的文档可能是库参考(Library Reference)。所有的文档都是可被搜索的。

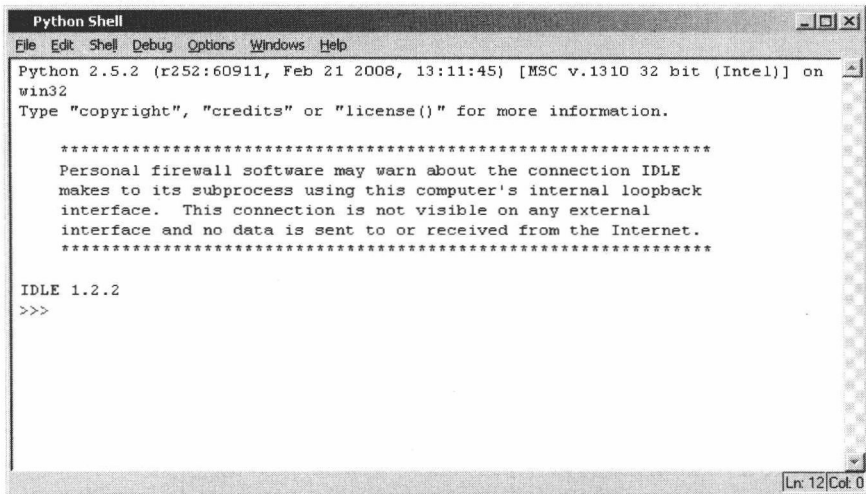


图1-1 IDLE交互式Python Shell

IDLE交互式Python Shell运行起来以后,你就可以接着阅读1.2节了。

^① 此菜单选项可能包括版本号,如Python 2.5。