

## <<Python高级编程>>

### 图书基本信息

书名：<<Python高级编程>>

13位ISBN编号：9787115217035

10位ISBN编号：7115217033

出版时间：2010-1-1

出版时间：人民邮电出版社

作者：Tarek Ziad é

页数：306

字数：422000

译者：姚军,夏海轮,译

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Python高级编程&gt;&gt;

## 前言

Python已经出现很长时间了。

曾几何时，我坚持使用Python，许多公司都认为我疯了。

现在，Python编码人员已经供不应求了。

诸如Google、YouTube、VMware和DreamWorks等重要的公司都在不断地争夺能找到的Python人才。

Python过去一贯落后于Perl，因为Perl拥有CPAN。

而现在，setuptools和PyPI已经引发了高可用的、高质量的第三方Python程序库的大爆发。

Python也曾经落后于JavaServlets和RubyOn：Rails，因为没有标准的用于与Web服务器交互的API。

现在，Web服务器网关接口（WSGI）引领了PythonWeb世界的复兴。

有了GoogleAppEngine，我想我们还将看到更多。

Python似乎对很固执并对简洁性有真正品味的编程人员具有吸引力。

很少有人因为学校里的学习任务或者大公司都在使用Python而成为Python编程人员。

人们只有在发现了Python的内在美才会沉迷于它。

因此，Python的书多得令人吃惊。

我没有足够的统计数字来证明，但是，似乎Python的编程书籍要多于其他语言。

然而，一直没有出现足够高级的：Python书籍，直至本书的出现。

本书介绍了一系列有趣的主题。

将介绍：Python的一组特性，以及以意想不到的方式使用它们的方法。

此外，还介绍了一组精选的、有趣的第三程序库和工具，以及使用Python工具和程序库的敏捷编程方法。

这包括基于nose的测试驱动开发，基于doctest的文档驱动开发，使用Mercurial进行源代码控制，使用Buildbot实现持续集成，以及使用Frac完成项目管理。

最后，介绍了一些更传统的主题，如剖析、优化以及诸如AlexMartelli的Borg方法，还介绍了诸如单例之类的设计模式。

如果你正打算从了解Python进步到精通Python，那么本书正适合你。

实际上，这正是5年前我所希望拥有的书。

我花费了数年，通过踏踏实实地参加：PyCon和本地的Python用户组而得到的一切，现在已经都在这一本简洁的书当中了。

没有什么比成为Python编程人员更激动人心的了！

## <<Python高级编程>>

### 内容概要

本书通过大量的实例，介绍了Python语言的最佳实践和敏捷开发方法，并涉及整个软件生命周期的高级主题，诸如持续集成、版本控制系统、包的发行和分发、开发模式、文档编写等。本书首先介绍如何设置最优的开发环境，然后以Python敏捷开发方法为线索，阐述如何将已被验证的面向对象原则应用到设计中。

这些内容为开发人员和项目管理人员提供了整个软件工程中的许多高级概念以及专家级的建议，其中有些内容的意义甚至超出了Python语言本身。

本书针对具备一定Python基础并希望通过在项目中应用最佳实践和新的开发技术来提升自己的Python开发人员。

## <<Python高级编程>>

### 作者简介

Tarek Ziad é , 是巴黎Ingeniweb公司的CTO, 其工作方向为Python、 Zope、 Plone技术和质量保证。

他参与Zope社区已经有5年了, 并且曾经为Zope自身贡献过代码。

Tarek创建了Afpv, 这是法国的Python用户组, 并且编写了两本法语的Python书籍。

他还在诸如Solutions Linux、 Pycon、 OSCON和EuroPython等法国及国际会议上发表了许多演讲, 并且主持了许多课程。

## &lt;&lt;Python高级编程&gt;&gt;

## 书籍目录

第1章 准备工作	1.1 安装Python	1.1.1 Python实现版本	1.1.2 在Linux环境下安装
	1.1.3 在Windows环境下安装	1.1.4 在Mac OS X环境下安装	1.2 Python命令行
	1.2.1 定制交互式命令行	1.2.2 iPython: 增强型命令行	1.3 安装setuptools
工作原理	1.3.2 使用EasyInstall安装setuptools	1.3.3 将MinGW整合到distutils中	1.3.1
工作环境	1.4.1 使用文本编辑器与辅助工具的组合	1.4.2 使用集成开发环境	1.4 小结
第2章 语法最佳实践——低于类级	2.1 列表推导	2.2 迭代器和生成器	2.2.1 生成器
器	2.2.2 协同程序	2.2.3 生成器表达式	2.2.4 itertools模块
	2.3.1 如何编写装饰器	2.3.2 参数检查	2.3 装饰器
文提供者	2.4 with和contextlib	2.4.1 contextlib模块	2.3.3 缓存
		2.4.2 上下文实例	2.3.4 代理
第3章 语法最佳实践——类级	3.1 子类化内建类型	3.2 访问超类中的方法	2.3.5 上下文
理解Python的方法解析顺序	3.2.2 super的缺陷	3.3 最佳实践	2.5 小结
3.4.1 描述符	3.4.2 属性	3.4 描述符和属性	3.2.1
3.4.2 描述符	3.5 槽	3.6 元编程	3.2.2
__metaclass__方法	3.7 小结	4.1 PEP 8和命名最佳实践	3.6.1 __new__方法
第4章 选择好的名称	4.1 PEP 8和命名最佳实践	4.2 命名风格	3.6.2
	4.2.1 变量	4.2.2 函数和方法	4.2 命名风格
	4.2.2 函数和方法	4.2.3 属性	4.2.1 变量
和包	4.3 命名指南	4.3.1 使用“has”或“is”前缀命名布尔元素	4.2.2 函数和方法
命名序列元素	4.3.3 用显式的名称命名字典	4.3.2 用复数形式	4.2.3 属性
名称	4.4 参数最佳实践	4.3.3 用显式的名称命名字典	4.2.4 类
	4.4.1 根据迭代设计构建参数	4.3.4 避免通用名称	4.2.5 模块
	4.4.2 信任参数和测试	4.3.5 避免现有	4.3 命名指南
	4.4.3 小心使用*args和**kw魔法参数	4.4 参数最佳实践	4.3.1 使用“has”或“is”前缀命名布尔元素
	4.5 类名	4.5 类名	4.3.2 用复数形式
	4.6 模块和包名称	4.6 模块和包名称	4.3.3 用显式的名称命名字典
	4.7 使用API	4.7 使用API	4.3.4 避免通用名称
	4.7.1 跟踪冗长	4.7.1 跟踪冗长	4.3.5 避免现有
	4.7.2 构建命名空间树	4.7.2 构建命名空间树	4.4 参数最佳实践
	4.7.3 分解代码	4.7.3 分解代码	4.4.1 根据迭代设计构建参数
	4.7.4 使用Egg	4.7.4 使用Egg	4.4.2 信任参数和测试
	4.7.5 使用deprecation过程	4.7.5 使用deprecation过程	4.4.3 小心使用*args和**kw魔法参数
	4.8 有用的工具	4.8 有用的工具	4.5 类名
	4.8.1 Pylint	4.8.1 Pylint	4.6 模块和包名称
	4.8.2 CloneDigger	4.8.2 CloneDigger	4.7 使用API
小结	第5章 编写一个包	第6章 编写一个应用程序	4.7.1 跟踪冗长
第9章 生命周期管理	第10章 编写项目文档	第7章 使用zc.buildout	4.7.2 构建命名空间树
和剖析技术	第11章 测试驱动开发	第8章 代码管理	4.7.3 分解代码
	第12章 优化: 通用原则	第9章 生命周期管理	4.7.4 使用Egg
	第13章 优化: 解决方案	第10章 编写项目文档	4.7.5 使用deprecation过程
	第14章 有用的设计模式	第11章 测试驱动开发	4.8 有用的工具
		第12章 优化: 通用原则	4.8.1 Pylint
		第13章 优化: 解决方案	4.8.2 CloneDigger
		第14章 有用的设计模式	4.9 小结

## <<Python高级编程>>

### 编辑推荐

Python专家TarekZiade将带你漫游Python应用程序开发的全过程。

《Python高级编程》首先介绍如何配置最优的开发环境，然后以Python敏捷开发方法为线索，告诉你如何将经过验证的面向对象原则应用到自己的设计中。

《Python高级编程》是关于Python最佳实践以及适用于Python应用程序的敏捷开发方法的权威说明，用实例展示了设计、编码和交付Python软件的实践过程。

你将从《Python高级编程》中学到以下技能：配置高效的开发环境；自定义Python提示符和用于部署的setuptools；编写高效语法iterator、generator和decorator；根据设计设置参数，遵循设计API的最佳实践；构建、发布和交付应用程序；编写基于多个egg的应用程序；使用zc.buildout交付、部署应用程序；构建并发布软件包，安排开发周期；使用版本管理系统和持续集成策略管理代码；实践测试驱动开发；对代码进行监测与优化。

提高软件运行速度；在应用程序中使用设计模式。

旧金山湾Python社区主持人 Shannon-jj Behrens倾情作序

<<Python高级编程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>