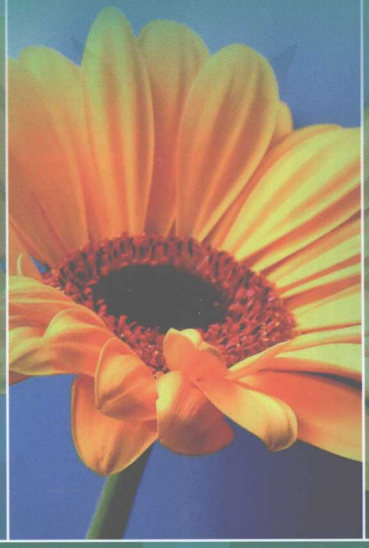
动态语言技术

精品书廊

Broadview
www.broadview.com.cn



Python源码剖析

——深度探索动态语言核心技术

陈儒 著

哲思自由软件社区 审校

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

Python 源码剖析

——深度探索动态语言核心技术

陈 儒 著

哲思自由软件社区 审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

作为主流的动态语言，Python 不仅简单易学、移植性好，而且拥有强大丰富的库的支持。此外，Python 强大的可扩展性，让开发人员既可以非常容易地利用 C/C++编写 Python 的扩展模块，还能将 Python 嵌入到 C/C++程序中，为自己的系统添加动态扩展和动态编程的能力。

为了更好地利用 Python 语言，无论是使用 Python 语言本身，还是将 Python 与 C/C++交互使用，深刻理解 Python 的运行原理都是非常重要的。本书以 CPython 为研究对象，在 C 代码一级，深入细致地剖析了 Python 的实现。书中不仅包括了对大量 Python 内置对象的剖析，更将大量的篇幅用于对 Python 虚拟机及 Python 高级特性的剖析。通过此书，读者能够透彻地理解 Python 中的一般表达式、控制结构、异常机制、类机制、多线程机制、模块的动态加载机制、内存管理机制等核心技术的运行原理，同时，本书所揭示的动态语言的核心技术对于理解其他动态语言，如 Javascript、Ruby 等也有较大的参考价值。

本书适合于 Python 程序员、动态语言爱好者、C 程序员阅读。

图书在版编目（CIP）数据

Python 源码剖析：深度探索动态语言核心技术 / 陈儒 著.—北京：电子工业出版社，2008.6
ISBN 978-7-121-06874-4
I. P… II.陈… III.软件工具－程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字（2008）第 083441 号

责任编辑：杨绣国
印　　刷：北京智力达印刷有限公司
装　　订：北京中新伟业印刷有限公司
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：787×980　1/16　印张：31.75　字数：600 千字
印　　次：2008 年 6 月第 1 次印刷
印　　数：4 000 册　　定价：69.80 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。

推荐序一

让我们做得更好

Python，我想已经不再是个陌生的词了，越来越多的人开始学习它，使用它，宣传它，甚至用它找到了工作。如果你了解 Python，那么我想问一下，你对它有多了解呢？它是一种什么语言？如何实现的？有哪些对象，它们是如何处理的？你了解 Python 的虚拟机吗？了解它的运行环境吗？其实作为初学者或只是使用者，你的确不必了解这么多细节的内容，但是探究事物的原理，分析底层细节却也是许多人成为高手、“老鸟”的原因，因为你知道别人不知道的东西，掌握了别人不了解的技术，这些内容使得你的见解、分析，甚至作品都可能超过别人。那么本书就向你提供了一个了解 Python 底层细节的机会，你可以沿着作者的思路和角度去体会 Python 的工作原理和底层的细节，一点一点地了解 Python 源码的精妙之处，有助于更好地掌握 Python 并编写出高质量的程序。

本书的内容深入到 Python 的方方面面，像 Python 的对象实现机制是如何用 C 来表现的；对象的特性是如何实现的；对象是如何管理的；不同对象，如 `int`、`str`、`dict`、`list` 等的处理；Python 的虚拟机框架、作用域的实现；运行时环境，pyc 文件，类机制等。还有一些高级话题，如内存管理，GIL（Global Interpreter Lock）与多线程，模块动态加载等。

在接触到本书之前，我已经在作者的 Blog 上见到过部分内容，那时已经被作者不懈的毅力和深厚的功力折服。说实话，由于经常接触 Python，对于原本熟悉的 C 语言也越来越陌生，更不要说去“啃”Python 的 C 代码了。而 Robert Chen 可以从源码中进行有条理的分析和整理，并终于出版此书。这不仅让人敬佩，更让广大的 Python 爱好者受益多多。因此，当出版社希望我为本书作一个推荐序时，我毫不犹豫地答应下来。

第一次见到 Robert Chen 还是在 CPUG 的一次会课上，那时 Robert Chen 给大家带来一个主题为“Python 作用域与名字空间”的讲座，让在座的 Pythoner 对 Python 的作用域机制有了更深入的理解，讲座效果非常好，讨论也很热烈。他从源码的角度讲述了 Python

的一些规则，使得大家的理解不再停留在形式上或规则上，而达到了本质或实现的层次，让我们“知其然，更知其所以然”。

如果你是一位热心的 Pythoner，想必会知道中文 Python 的邮件列表（*http://groups.google.com/group/python-cn*），从上面对有些问题的回复中，你会发现 Robert Chen 总是从源码及字节码实现的角度来回答问题，非常有说服力。因此当许多 Python 爱好者得知 Robert Chen 将出版此书时，都非常盼望，现在这本书终于出版，大家都深感庆幸！

本书不仅仅是高水平、高质量的一本书，纵观国内外与 Python 相关的书籍，它也是第一本从源码角度写作的书，所以意义非常。目前国内原创的 Python 书籍还不多，就我所知国内已经出版的几本 Python 方面的书尚不能满足读者需求，而本书应该不会让你失望。

不过本书应该不是面向初学者的书，因为它涉及了许多较深的内容和知识，建议读者应先掌握像 C 语言、数据结构、操作系统、编译原理等方面的基础知识，并且具备一定的编程经验，才能更好地理解书的内容。

再次感谢 Robert Chen 带给大家的这份礼物！

李迎辉

limodou@gmail.com

2008 年春

推荐序二

真的难以想象，Python 语言和社区能够发展得如此迅速。在我第一次使用 Python 完成我的项目时，它还不过是一个刚刚在开源社区中起步的新生儿，然后在各方面迅速推广，4 月 8 日，google 发布的 App Engine 更是让所有的开发人员眼中一亮。相信今后会有更多的开发人员投入到 Python 的技术领域中来。

记得在 2002 年时，我使用 Python 写了一套大规模的消息系统，几位同事分别使用 Python、Java 和 C 完成了一个异步二进制消息流的客户端和服务器。通过一系列测试，大家惊奇地发现 Python 以每秒一倍的数据处理量超过了 C 写的代码。后来，我的同事细心地查看了 Python 的源代码，发现了几种完全不同的操作系统调用方法，以及为提高性能而使用的技巧。这也是我第一次开始查看 Python 的源代码。最近的一次则是我在 xBayTable 中使用 asyncore 时，通过阅读 asyncore 的源代码排除了一个痛苦 bug，轻松地找到了问题的根源，很快就换了一种解决方案来继续我的工作。了解 Python 的源代码，我们能获得很多的好处：

- 使用 Python 方法提高自己的代码性能和功能；
- 快速地与文档结合，解决问题或是找出方法；
- 扩展 Python。

我常将所有的书分为口袋书、马桶书、枕头书。RobertChen 的《Python 源码剖析》，更多讲述的是 CPython 中的实现技术和方法。这可以让我们从不同的层面了解 Python 简洁背后的机理。我更推荐大家把它当做口袋书，在准备书写 Python 扩展前把它作为一本工具书，配合“Extending and Embedding the Python Interpreter”会让你更容易地完成你的工作。另一方面，当你想使用 Python 这种方法解决问题时，这本书也可以成为你的好伙伴，它让你更多更快地了解 Python 是怎么做的，从而做得“和 Python 一样”。当你对 Python 的一些问题百思不得其解时，这本书也许可以从不同的方面帮助你了解它最底层发生的故事。

最后，作为 Python 社区的长期参与者，希望更多的代码人不但能使用 Python 语言去完成自己的工作，也希望能有更多的人通过这本书成为 Python 语言的开发者，更希望中国有越来越多的 Python 开发者。

黄冬
2008 年 4 月于北京
下雨的深夜

推荐序三

非常高兴看到又一本原创的 Python 图书出版。

说起来，我和 Python 还算有一点缘分。在 2000 年的时候，一次非常偶然的机会我接触到 Python，当时网上的资料非常少，不知天高地厚的我竟冒失地接手了国内第一本引进版 Python 图书的合作翻译工作，往事不堪回首。记得当时经常有人问我 Python 能用来做什么……而我能举出来的例子的确寥寥可数。

历经数年的发展，Python 已今非昔比，各领域都不乏 Python 的成功案例。就拿 Web 方面来说，正如 PHP 给 Yahoo!带来的巨大动力，Python 在新一代的互联网霸主——Google 内部早就充当了重要角色，成为排名第三的“官方语言”。而就在前一段时间，Google 革命性的 App Engine 产品一经推出立即引起了莫大关注，其首选开发语言就是 Python。

纵观国内技术环境，Python 语言仍处于慢热的状态，应用仍然不算广泛。不过我们也已经有称得上比较成功的实现案例了，比如著名的 Web 2.0 的代表站点——豆瓣网即是用 Python 开发的，创始人杨勃对 Python 的效率和优雅赞誉有加。

Python 也是权威机构 TIOBE 评出的 2007 年度编程语言，这些“利好”的消息也将进而带动新一轮的技术走向，预示着 Python 更大规模的流行时代即将到来。

话说回来，“开放平台”在未来几年一定是个不可避免的技术趋势，而跟着大厂商的平台亦步亦趋，照猫画虎，想必也能开发出来繁多的周边应用，但开放未必对所有人都是个好事情，久而久之开发者难免有盲人摸象之感，很难掌握全局，掌握关键架构技术，故深入研究 Python 的基础技术仍不可少。这本《Python 源码剖析》的出版恰是好时机，弥补了国内图书在这方面的空白，此外，作者在 Python 领域的精耕细作的研究精神亦值得学习。

研读、分析源代码乃是提高编程技能的一条捷径，庖丁解牛方能游刃有余，夯实基础方可构建高楼大厦。

读这本《Python 源码剖析》就像一次探险之旅，祝愿朋友们能够获得一次愉悦的阅读体验。

冯大辉

2008 年 4 月于杭州

前言

缘起

第一次接触 Python，是通过《程序员》杂志上“恶魔吹着笛子来”的系列文章——《自由与繁荣的国度》。但是真正开始使用 Python，还是在进入实验室，开始研究自然语言处理和信息检索之后。自然语言处理其实大部分的时间都在与文本打交道，需要进行大量的对文本分析、统计的工作。开始的时候，我使用的是 C++，因为大学的时候第一门编程语言课就是 C，其后转向 C++是很自然的迁移。那时候觉得 C++很有一种高贵的感觉，因为 C++足够复杂，有足够多的 trick，尤其是像模板和泛型编程这样的新鲜玩意儿。掌握这么复杂的东西，也就意味着你的脑袋跟这东西一样复杂，这是很能让人虚荣的一件事。

C++的复杂性是个仁者见仁，智者见智的话题，但其实回到文本处理这个话题上来说，C++的 STL 在很大程度上已经足够好用了。文本处理不是服务器，所以不需要考虑自己管理内存，不需要考虑这个模式那个模式，STL 提供了足够多的工具，简单组装一下就可以用了。

但俗话说得好，“不怕不识货，就怕货比货”，当我开始尝试用 Python 来进行日常的工作之后，突然发现 C++太复杂了。对于 Python，我的感觉只有四个字：摧枯拉朽。我只需要简单地写一个 `l = [ ]`，再也不用写诸如 `list< map<string, string> > l = list< map<string, string> >()` 这样折磨人眼的东西了，这使得代码量急剧减少。对于采用 Python 这样的英明决策，我想，最满意的就是我的手指头了。

随着对 Python 的逐渐熟悉，我越来越惊叹于 Python 简洁的表达，强大的功能。尤其是 Python 表现出来的强烈的动态性。比如下面这段与解释器的交互过程：

```
>>> class A:
...     pass
...
>>> a = A()
>>> a.show()
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: A instance has no attribute 'show'
>>>
>>> class B:
...     def show(self):
...             print 'i am B'
...
>>> A.__bases__ = (B,)
>>> a.show()
i am B
>>>
```

这样的动态能力，在当时简直让我目瞪口呆。从那时候起，我就有了一个强烈的好奇心：Python 是怎么实现的？我们知道 Python 是用 C 语言实现的，那这些神奇的动态能力是怎么通过 C 语言完成的呢？

于是我开始上网搜索资料，然而我发现，详细介绍动态语言实现原理的资料根本就没有，只有一些零散的信息散落在各种资料中。再具体到 Python，唯一一篇与 Python 实现相关的资料是《The Architecture of Python》，这是美国一所大学的学生在一次课程设计中产生的文档。这份文档篇幅太短，内容也太简略了，只包含了一些最简单的信息，即使对作为 Python 中对象模型关键的 PyObject 结构体，也仅仅有一些简单的描述。最要命的是，其研究的对象是 Python 2.1，版本太久远了。而我准备开始研究 Python 如何实现时，Python 已经进化到了 2.4，Python 的对象模型已经发生了重大的改变，所以这份文档对于想要深入掌握 Python 的实现来说，几乎没有太大作用，让人有一种“食之无味、弃之可惜”的感觉（这份文档目前在 Google 上已经搜索不到了）。但不管怎么说，这份文档给了我对于 Python 实现的一个最初的认识，给了我一个起始点。

2004 年年末，我开始了探索 Python 如何实现的漫漫长征。我选择了编译这个最初的切入点，但是很快我就发现，Python 的编译过程中大量地使用了 Python 中的一些内置对象，所以我将切入点转向了 Python 的对象模型。在完成了 Python 对象模型和内建对象的剖析后，又重新转回到编译过程的剖析，我发现 Python 的编译过程实际上就是一个标准的编译过程，在任何一本关于编译原理的书上，你都可以找到它的实现过程。于是，我做了一个决定，不再剖析 Python 的编译过程，而是以 Python 的编译结果为起点，开始 Python 虚拟机的剖析，正因为这样，你在本书中看不到对 Python 的编译过程的剖析，不过别着急，你能够看到 Python 的编译结果，对于理解 Python 虚拟机的实现来说，这个编译结果才是最重要的。

很快，第一篇关于 PyObject 的笔记出炉了，紧接着，Python 中最简单的对象——整数对象的笔记也诞生了。到了这个时候，我开始对完成这项工作有信心了，虽然后来的经历证明我当时的信心是多么的虚妄。也是在这个时候，我开始看到 Python 实现的精妙之处，完成这项工作的兴趣和动力也越来越强。

随着对 Python 挖掘的不断深入，我开始碰到一个又一个的瓶颈，虚拟机的框架、函数的实现、class 机制、module 的动态加载……在没有太多外部资料的情况下，要想通过源码弄明白一样东西，实在是太艰难了。有的时候，由于难度太大，我甚至中断了剖析的工作，直到很久以后，才在好奇心和兴趣的强烈驱使下重新开始。正是由于这样的磕磕绊绊，到了 2007 年底，这项工作才算完成。花了两年多的时间，除了难度方面的原因，另一个原因是 2006 年底，Python 社区发布了 Python 2.5，为了保证与最新的 Python 实现保持一致，我将所有的剖析跟 Python 2.5 又对照了一遍，所以，你手上这本书，跟最新的 Python 实现是一致的。

谁需要这本书

什么人需要这本书呢？呃，当然，我自然是希望需要这本书的人越多越好，我巴不得所有的程序员人手一本☺。从我自身的体会来说，我觉得以下三类人都会对这本书感兴趣：Python 程序员、动态语言爱好者、C 程序员。

Python 程序员当然能从这本书得到最多的收获，在这本书里，你能找到 Python 所有魔法的最终来源，每一个看似神奇的特性都会得到一个最朴素的解释，每一个所谓的高级特性，比如 decorator，都会变成你手中平凡的工具。当然，你还会对你编写的 Python 代码有一个最透彻的认识。比如下面这段代码，为什么不能加载 D 呢？

```
[A.py]
from B import D

class C:
  pass

[B.py]
from A import C

class D:
  pass
```

当最终完成对 Python 的剖析之后，我发现对于 Python 代码，可以有一种洞彻肺腑的感觉。看着代码，在脑海里就完全可以出现虚拟机在后台如何运作的情景，这种感觉非常棒。

除了 Python 程序员，其他的动态语言爱好者都能从这本书得到很多有益的信息。不论是 Python 在其他平台的实现——IronPython、Jython，还是其他的动态语言——Ruby、Javascript，只要你对动态语言感兴趣，想要了解动态语言的实现机理，那赶紧把这本书搬回家。动态语言在大致的架构和实现机制上，有很多的相似之处。

这本书在本质上是探讨如何用 C 语言实现 Python 这门动态语言的，所以 C 程序员能从中获得巨大的收益。作为一个优秀的开源项目，通过阅读它的源码，C 程序员可以从中汲取世界上最优秀的 C 程序员的经验。当然，对于另一部分特殊的 C 程序员，这本书的价值就更大了，如果你需要为 Python 编写 C 扩展模块，或者更酷地，你想要在你的系统中嵌入一个脚本引擎，就像很多网络游戏中的脚本引擎，那么本书就是你必备的了。

尽管这本书看上去讨论的是个高深的话题，但其实对读者知识储备的要求相当少。只需要 C 语言和 Python 语言的基础知识。

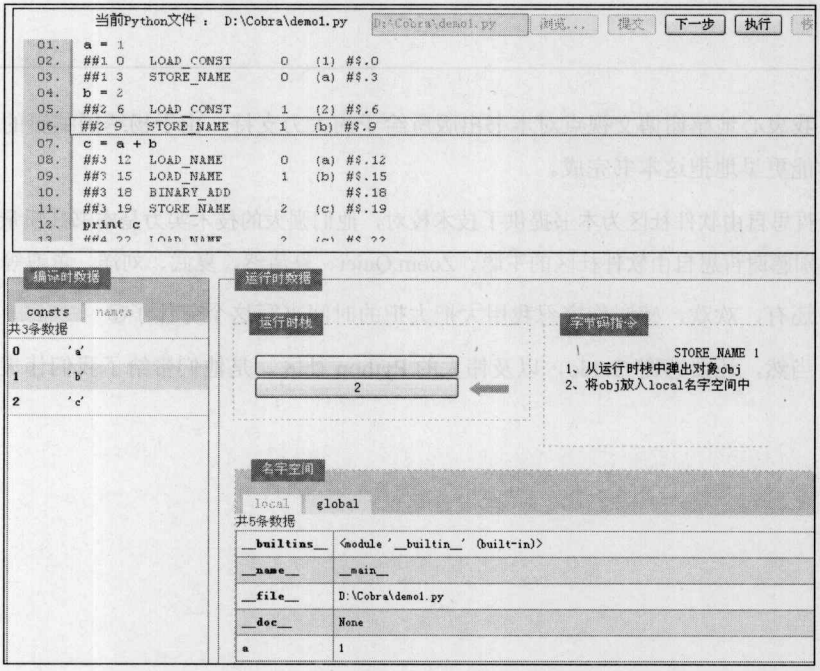
1．C 语言的基础知识是必须的，但是不必多么精通 C，只需要能看懂 C 代码就可以。

2．Python 语言的基础知识也是必须的，这更不是问题了，有编程基础的读者，半天的时间，就能看懂所有的 Python 语法，这本书基本上不涉及任何 Python 的库（在线程部分，有一些线程库的介绍），所以，你只需要熟悉基本的 Python 语法就可以了。

如何阅读这本书

关于这个问题，我的建议是，不要只阅读，更要实践。最基本的，你应该打开自己最熟悉的代码浏览工具，在真实的源码和书中的描述之间比对揣摩。更进一步，你可以随着本书的进展，动手捣鼓捣鼓 Python 的源码，用真实的输出验证书中的描述和你的理解。对于技术这个东西，再生花的妙笔也不能让你仅仅通过“阅读”便能乾坤在握，唯有亲身尝试，才能深解其中三昧。

为了帮助读者更好地利用此书，我在 Google Code 上发起了一个旨在可视化 Python 虚拟机的开源项目——Cobra（*http://code.google.com/p/python-cobra/*），其目的在于将 Python 虚拟机在执行一条条字节码指令时的运行时环境，以及虚拟机的状态变化，以可视的形式展现出来，以更加生动形象的方式加深读者对 Python 虚拟机的理解。同时，我也期望这个项目能成为有兴趣的读者锻炼自己改造 Python 虚拟机的能力的平台。该项目还处于发展初期，目前仅仅实现了一般表达式的可视化，我会尽快增强 Cobra 的功能，也希望感兴趣的读者一起加入到这个有趣的项目中来。下图是目前 Cobra 对简单的一般表达式的可视化效果。



代码下载

本书还提供了书中涉及到的两个简单项目的源代码，一个是对 Python 的最简化模拟的 Small Python，另一个是对 pyc 文件进行解析的解析器。两个项目的代码都可以在博文对本书的支持网站下载：*http://bv.csdn.net/resource/pythonympx.rar*。

联系作者

面对如此复杂而又伟大的 Python，以一本书的篇幅，难免对有的主题的涉及会显得简略；另一方面，以一人之力，在剖析与撰写中也难免会有错误与遗漏。如果你有什么建议，或者认为我还遗漏了什么东西，非常期待与你交流。你可以通过 *search.pythoner@gmail. com* 与我直接联系；或者，如果你愿意结识更多的 Python 同道，你也可以访问中文 Python 用户组（*https://groups.google.com/group/python-cn*），把你的心得与中国的 Python 爱好者一起分享，我也会一直在那里对本书进行支持。

致谢

我衷心地感谢博文视点对本书出版所给予的大力支持，由于杨绣国编辑的大力推动，我才能更早地把这本书完成。

哲思自由软件社区为本书提供了技术校对，他们强大的技术实力是本书高质量的保证，在此特别感谢哲思自由软件社区的王聪、Zoom.Quiet、夏清然、夏武、刘洋、董溥等。

还有，欢欢，感谢你容忍我用大把大把的时间来写这个劳什子☺

当然，还要感谢 Guido，以及伟大的 Python 社区，是他们带给了我们伟大的 Python。

联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱：*reader@broadview.com.cn*

投稿信箱：*bvtougao@gmail.com*

北京博文视点资讯有限公司（武汉分部）

湖北省武汉市洪山区吴家湾邮科院路特 1 号湖北信息产业科技大厦 1402 室

邮政编码：430074

电　　话：（027）87690813

传　　真：（027）87690595

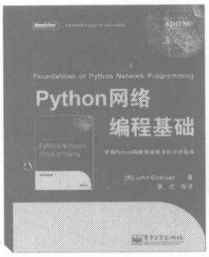
若您希望参加博文视点的有奖读者调查，或对写作和翻译感兴趣，欢迎您访问：

http://bv.csdn.net

关于本书的勘误、资源下载及博文视点的最新书讯，欢迎您访问博文视点官方博客：

http://blog.csdn.net/bvbook

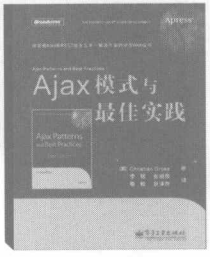
博文视点Web开发系列



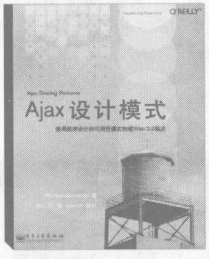
《Python网络编程基础》
John Goerzen 著
莫迟 等译
- Python网络编程最佳入门图书
- FTP、Email、XML、Web Service、多线程、异步通信……
  完整涵盖网络编程的方方面面

《构建可扩展的Web站点》
Cal Henderson 著
徐宁 译
- Web 2.0代表网站Flickr总架构师
  Cal Henderson经典力作

《Web开发人员参考大全》
Lázaro Issi Cohen,Joseph Issi Cohen 著
胡为君 译
- 没有你查不到的属性
- 没有你找不到的标签
- 没有你用不到的知识

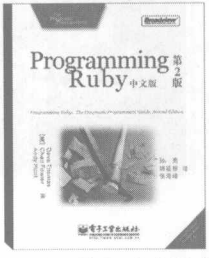


《Ajax模式与最佳实践》
Christian Gross 著
李锟 等译
- 深邃洞察Web开发整体架构
- 全面涵盖客户端与服务器端开发



《Ajax设计模式》
Michael Mahemoff 著
杨仁和 译 Ajaxcn 审校
- 全景展现Ajax技术多年智慧结晶
- 全面总结Web 2.0开发技术架构

《AJAX HACKS中文版：
创建快速响应Web站点的工具和技巧》
Bruce W. Perry 著
张险峰 等译
- 80个AJAX独门绝技一网打尽



《Programming Ruby中文版，第2版》
Dave Thomas,Chad Fowler, Andy Hunt 著
孙勇 姚延栋 张海峰 译
- 全球公认Ruby权威入门参考指南
- 让编程成为你的直觉
- 让开发成为你的乐趣

《Web开发敏捷之道
——应用Rails进行敏捷Web开发，第2版》
Dave Thomas, David Heinemeier Hansson 著
林芷薰 译 透明 审校
- 70%全新内容
- Ruby On Rails 经典力作最新第2版
- 第1版荣获Jolt Award（震撼大奖）

《PHP程序设计（第2版）》
Rasmus Lerdorf, Kevin Tatroe 等著
陈浩 胡丹 徐景 译
- PHP 5的权威指南书籍
- 包含PHP创始人Rasmus Lerdorf等PHP专家的独特见解
- 中国四大权威PHP社区鼎力推荐