

TURING

图灵程序设计丛书

Ruby

基础教程

第4版

[日] 高桥征义 后藤裕藏 著 [日] 松本行弘 审校 何文斯 译

Programming



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Ruby

基础教程

第4版



[日] 高桥征义 后藤裕藏 著 [日] 松本行弘 审校 何文斯 译

Programming

人民邮电出版社
北 京

图书在版编目(CIP)数据

Ruby 基础教程: 第4版 / (日) 高桥征义, (日) 后藤裕藏著; 何文斯译. -- 北京: 人民邮电出版社, 2014.9

(图灵程序设计丛书)

ISBN 978-7-115-36646-7

I. ①R… II. ①高… ②后… ③何… III. ①计算机网络—程序设计—教材 IV. ①TP393.09

中国版本图书馆CIP数据核字(2014)第167290号

TANOSHII RUBY, THE 4TH EDITION

Copyright © 2013 MASAYOSHI TAKAHASHI, YUUZOU GOTOU

Supervised by YUKIHIRO MATSUMOTO

All rights reserved.

Originally published in Japan by SB Creative Corp.

Chinese (in simplified characters only) translation rights arranged with

SB Creative Corp., Japan through CREEK&RIVER Co., Ltd.

本书中文简体字版由 SB Creative Corp. 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。

本书为日本公认的最好的 Ruby 入门教程。松本行弘亲自作序并推荐。本书对应最新的 Ruby 2.0, 也附带讲解了可运行于 1.9 版本的代码。事无巨细且通俗易懂地讲解了编写程序时所需要的变量、常量、方法、类、流程控制等的语法, 以及主要类的使用方法和简单的应用, 让没有编程经验的读者也能轻松掌握 Ruby, 找到属于自己的快乐编程方法, 做到融会贯通, 灵活运用到实际工作中。

本书适合 Ruby 初学者学习参考, 有一定 Ruby 基础的读者若想再回顾一下 Ruby 的各知识点, 本书也能提供不少帮助。

◆ 著 [日] 高桥征义 后藤裕藏

译 何文斯

责任编辑 徐 骞

加工编辑 杜晓静 徐 骞

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>

北京圣夫亚美印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 22

字数: 520千字

2014年9月第1版

印数: 1-4 000册

2014年9月北京第1次印刷

著作权合同登记号 图字: 01-2013-9264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

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

推荐序

Ruby 常常被称为“国产语言”。作为 Ruby 的设计者，我的确是个如假包换的日本人，Ruby 最具代表性的实现——CRuby^① 中的许多核心成员也都是日本人。但是，对 Ruby 的开发和发展做出过大大小小贡献的人里有很多都不是日本人。在 JVM 上使用的 JRuby、用 C++ 实现的 Rubinius 等，其主要开发者是美国人；MacRuby、RubyMotion 的主要开发者是比利时人。还有使 Ruby 发扬光大的 Ruby 社区，其大部分活动都在日本以外的国家或地区进行。Ruby 社区里最具代表性，也是最早的 Ruby 技术大会——RubyConf 每年都会在美国举行。除此以外，在美洲、欧洲、亚洲等世界各地也都会举办其他的 Ruby 技术大会。Ruby 是无数人努力的结晶，是一个社区，是一种文化。所以，我多少有点反感因 Ruby 诞生于日本就将其冠以“国产”的说法。

但是，日本也有引以自豪之处。一是日本拥有世界上最早建立的 Ruby 社区。我访问过许多国家的社区，可以说日本的社区是世界上水平最高的。另外，日本拥有一批经验丰富的 Ruby 社区成员。他们是 Ruby 最早的一批使用者，并通过各种活动和实际开发，孕育出了属于 Ruby 自身的多元文化。这样的人才才是 Ruby 社区中可贵的瑰宝。

本书是最早使用 Ruby 的先驱者们为了欢迎下一批社区成员而写的一本入门书。本书前 3 个版本帮助过无数新人融入到 Ruby 社区。这次，对应 Ruby 最新版的第 4 版比以往更详细，更通俗易懂，对大家学习 Ruby 会有很大的帮助。通过学习本书，衷心希望大家体会到 Ruby 编程带来的乐趣。

2013 年 4 月

松本行弘

^① 也称 MRI。——译者注

译者序

曾经有同事问我，为什么这么喜欢 Ruby？我的回答是，因为 Ruby 非常有趣，用 Ruby 写程序是一件快乐的事情。对方满脸困惑，似乎在质疑——写程序也能让人感到快乐？的确，现在不少人认为编程是一件又苦又累的差事。代码搬运工、码农等大家的自嘲语也很难让人把编程与快乐联系在一起。回想当初刚学习编程的时候，我们曾因为实现了某个算法、某个功能而感到兴奋，而工作后却被项目进度、加班等压得喘不过气来，似乎已经忘记了编程原本是一件令人快乐的事情。

“快乐编程”是本书的主旨，也是 Ruby 令人着迷的原因之一。本书继承了日语技术类书籍的优良传统，采用了大量图、表、例子，讲解通俗易懂。从编程基础的数据类型、控制语句，到面向对象编程、鸭子类型、正则表达式等高级编程技巧，带领着读者逐步进入 Ruby 的程序世界，使大家沉浸在编程的乐趣之中。而对于久经沙场的“老鸟”们，Ruby 那如诗篇一样优雅的语法、各种魔术般的语法糖，以及能把我们从枯燥无味的重复劳动中解放出来的丰富强大的类库，都一定都能唤起大家的“集体回忆”，重拾已经失去的编程乐趣。

2007 年接触 Ruby 后，我就喜欢上了这个“小家伙”。偶然一次机会，我从 Ruby China 社区得知图灵公司正在寻找这本书的译者。非常幸运，我得到了这个宝贵的机会。

在此非常感谢图灵公司以及 Ruby China 社区，也非常感谢翻译过程中图灵公司各位编辑给予的帮助。

这是我第一次译书，其间所耗费的时间与精力远远超出了当初的预期。翻译期间，我牺牲了很多与家人共处的时间，在此深深感谢家人们的谅解、关心与支持，同时也非常感谢朋友们、同事们在这段日子里给我的鼓励与支持。

参与本书的翻译，是我人生中一次奇妙的经历。记得以前我曾经对计算机硬件非常着迷，经常阅读硬件杂志。记得当时有一本计算机硬件入门杂志，整本都是采用彩色铜版纸

印刷，图文并茂，手把手地教读者装配、使用计算机。不过慢慢地，有读者抱怨内容太浅显，希望作者能写点高深的内容。当时杂志编辑的一段回复，到现在我还记忆犹新，大意是“我们的任务就是迎接更多的新朋友，同时让更多的老朋友抛弃我们，当你觉得我们已经无法满足你的求知欲时，那么恭喜你，你已经毕业了，我们的任务也完成了”。这也是我此刻的心情。

最后，预祝大家通过本书都能找到属于自己的 Ruby 快乐编程之道。

何文斯

2014 年 5 月 4 日，写于广州

前言

乐在其中的编程语言

与计算机程序“交流”的方式有两种。第一种方式是使用程序，另外一种编写程序。

然而，编写程序的人相对要少，大部分人都是使用程序而已。这个有点接近“读文章的人”与“写文章的人”的比例。读小说、散文、纪实文学等的人很多，但写小说、纪实文学的人数量上就远比读者少。

这里说的“文章”不仅仅是指商业出版物，还包括个人网站。有很多人几乎每天都更新博客，有的是与身边的人分享有趣的事情，有的是提供某些有用的信息。虽然可能只是一些微不足道的信息，但还是会有读者乐于阅读，这类读者就是“用户”，因此博客也可以说是一种“供读者阅读的文章”。

大家基于各种目的创建了这类网站，其中不少人是因为很享受自己编写内容的过程。以个人网站为例，单纯追求创作乐趣的人可能会更多。

编程不也是如此吗？也就是说，并不仅仅是为了某种目的而编程，而是因为编程时乐在其中。

编程的乐趣并非单指程序内容，使用的编程语言不同，所获得的乐趣也不一样。像这样，让编程本身变得有趣的编程语言真的存在吗？

——存在。Ruby 就是其中一种。

* * *

Ruby 是一种旨在使大家编程时能乐在其中的编程语言。完全面向对象，有丰富的类库，直观、人性化的语法等都是 Ruby 的特征，但这些并不是 Ruby 的目的，只是快乐编程的手段。

在程序世界里，有着种类繁多的语言。这些语言诞生的缘由多种多样，有的是为了编写运行速度快的程序，有的是为了可以在短时间内编写程序，有的是为了让程序只需编写一次就可以在任何环境中运行，有的是为了使小孩也能进行简单编程，等等。但是，似乎并没有哪个语言积极地宣称其目的是为了快乐编程。这可能是由于各个语言的设计者，并没有认真考虑过让任何人都可以编程。

当然，使大家编程时乐在其中的语言，肯定是一种简单易掌握的语言，复杂的语言不可能让人体会到快乐。同时，这门语言又必须是一个功能强大的语言，若非如此，实际编写程序时会非

常费劲。毋庸置疑，Ruby 就是这样一种简单易掌握，并且功能强大的编程语言。

* * *

为了让零编程经验的读者轻松掌握 Ruby，本书会巨细无遗地介绍 Ruby。从编写程序时所需要的变量、常量、方法、类、流程控制等的语法说明，到主要类的使用方法和简单的应用，都会尽量用通俗易懂的方式来说明。对于从未接触过计算机的读者来说，也许这有点难，但是那些稍懂点 HTML 的读者很容易就能做到融会贯通。另外，对于那些并非初学者的读者来说，若想再回顾一下 Ruby 的各知识点，本书也能提供不少帮助。

希望各位读者能通过本书，熟练掌握 Ruby，找到属于自己的快乐而有趣的编程方式，并灵活运用到实际中，笔者将不胜荣幸。

欢迎来到 Ruby 的世界！

高桥征义 | 后藤裕藏

0.1 关于 Ruby

在开始编程之前，让我们先了解一下什么是 Ruby。

◎ Ruby 是脚本语言

用 C 或者 Java 语言编写的程序，在运行前需要执行编译这一步骤，把源码翻译成计算机可以理解的机器码。而用脚本语言编写的源码并不需要编译，直接运行程序便可。

也就是说，在使用脚本语言时，开发流程会从

源码编写 → 源码编译 → 程序运行

变为

源码编写 → 程序运行

因此，与需要编译的语言相比，Ruby 更能让大家轻松享受到编程之趣。

◎ Ruby 是面向对象的语言

Ruby 是一群热爱面向对象编程的程序员，为了实现最优秀的面向对象语言而设计、开发的一门语言。它是完全面向对象的，所思考的东西都可以直接通过代码表达出来。^①同时，Ruby 也具有继承、Mix-in 等面向对象语言的必备特性。

另外，Ruby 不仅提供了丰富的标准类库，还具有对应各种异常的错误处理机制、自动释放内存的垃圾回收机制等提高编程效率的特性。

◎ Ruby 是跨平台的语言

Ruby 能在 Mac OS X、Linux、FreeBSD、Solaris 等类 Unix 操作系统以及 Windows 操作系统等平台上运行。它的大部分脚本无需修改即可在各个不同的平台环境下运行。

◎ Ruby 是开源软件

Ruby 诞生时，松本行弘先生就公开了源码，使之成为开源软件（自由软件，Free Software）。任何人都可以随意获取 Ruby，并自由使用。自 1995 年松本行弘先生在互联网上发布 Ruby 以来，Ruby 得到了来自各方的广泛支持，并一直活跃至今。

0.2 本书的读者对象

本书是一本入门级图书，面向具备一定计算机知识但没有编程经验的读者，旨在帮助他们掌握 Ruby 编程知识。本书尽量以无需具备专业知识也能读懂的方式向大家介绍 Ruby，但省略了“启动 / 关闭计算机”“Shift 键的使用方法”等基础知识的说明。本书面向这样的读者：

^① 与面向过程的编程方法相比，我们一般认为面向对象的编程方法比较符合人的思维习惯。——译者注

- 具备操作文件和执行命令等基础的计算机知识
- 可使用编辑器创建文本文件
- 计划学习编程

0.3 本书的构成

本书采用“循序渐进，逐步深入”的写作方式，对于有 Ruby 基础的读者来说，前半部分或许有些沉闷。建议已经掌握 Ruby 语法等基础知识的读者，快速浏览前两部分，从第 3 部分开始仔细阅读。

◎ 第 1 部分 Ruby 初体验

利用简单的 Ruby 小程序，介绍计算机程序的基本构成。

◎ 第 2 部分 Ruby 的基础

介绍 Ruby 语法、规则等 Ruby 编程基础知识，以及类、模块等面向对象编程的思考方法和术语。

◎ 第 3 部分 Ruby 的类

要编写程序，只懂语法还远远不够。Ruby 之所以能使大家快乐编程，主要缘于 Ruby 精心设计的标准类库。

在本部分，我们会列举多个 Ruby 的基础类，介绍其功能和使用方法。

◎ 第 4 部分 动手制作工具

在本部分，我们将进行一次总复习，介绍一些稍微复杂点的 Ruby 程序，让大家尝试一下如何用 Ruby 编写实际的程序。

◎ 附录 A Ruby 运行环境的构建

介绍各个平台的 Ruby 安装方法。

◎ 附录 B 参考

介绍使用 Ruby 时所需的知识以及各相关信息。

0.4 Ruby 的运行环境

本书内容适用于 Ruby 2.0.0 版本，并兼容 Ruby 1.9 系列版本。适用的操作系统为 Windows 7/8 和 Mac OS X、Linux 等常见类 Unix 操作系统。

在继续阅读本书前，请读者按照附录 A.1 节的说明，安装 Ruby 运行环境。

目录

第 1 部分 Ruby 初体验

第 1 章 Ruby 初探	2	1.13.2 times 方法	18
1.1 Ruby 的运行方法	3	第 2 章 便利的对象	19
1.1.1 ruby 命令的执行方法	3	2.1 数组	20
1.1.2 irb 命令的执行方法	4	2.1.1 数组的创建	20
1.2 程序解说	5	2.1.2 数组对象	20
1.2.1 对象	5	2.1.3 从数组中抽取对象	20
1.2.2 方法	5	2.1.4 将对象保存到数组中	21
1.3 字符串	6	2.1.5 数组的元素	22
1.3.1 换行符与 \	6	2.1.6 数组的大小	22
1.3.2 ' ' 与 " "	7	2.1.7 数组的循环	23
1.4 方法的调用	7	2.2 散列	24
1.5 puts 方法	8	2.2.1 什么是符号	24
1.6 p 方法	8	2.2.2 散列的创建	25
1.7 中文的输出	9	2.2.3 散列的使用	25
1.8 数值表示与计算	11	2.2.4 散列的循环	25
1.8.1 数值	11	2.3 正则表达式	26
1.8.2 四则运算	11	模式与匹配	26
1.8.3 数学相关的函数	12	第 3 章 创建命令	29
1.9 变量	13	3.1 命令行的输入数据	29
print 方法和变量	14	3.2 文件的读取	30
1.10 注释	14	3.2.1 从文件中读取内容并输出	31
1.11 控制语句	15	3.2.2 从文件中逐行读取内容并输出	32
print 方法和变量	16	3.2.3 从文件中读取指定模式的内容 并输出	33
1.12 条件判断: if ~ then ~ end	16	3.3 方法的定义	34
1.13 循环	18	3.4 其他文件的引用	34
1.13.1 while 语句	18		

8.3	扩展类	99	9.6	定义运算符	122
8.3.1	在原有类的基础上添加方法	99	9.6.1	二元运算符	122
8.3.2	继承	99	9.6.2	一元运算符	124
8.4	alias 与 undef	101	9.6.3	下标方法	125
8.4.1	alias	101	第 10 章 错误处理与异常	126	
8.4.2	undef	102	10.1	关于错误处理	126
8.5	模块是什么	103	10.2	异常处理	127
8.6	模块的使用方法	103	10.3	异常处理的写法	128
8.6.1	提供命名空间	103	10.4	后处理	131
8.6.2	利用 Mix-in 扩展功能	104	10.5	重试	131
8.7	创建模块	104	10.6	rescue 修饰符	132
8.7.1	常量	105	10.7	异常处理语法的补充	132
8.7.2	方法的定义	105	10.8	指定需要捕捉的异常	133
8.8	Mix-in	106	10.9	异常类	134
8.8.1	查找方法的规则	107	10.10	主动抛出异常	135
8.8.2	extend 方法	109	第 11 章 块	136	
8.8.3	类与 Mix-in	110	11.1	块是什么	136
8.9	面向对象程序设计	111	11.2	块的使用方法	137
8.9.1	对象是什么	111	11.2.1	循环	137
8.9.2	面向对象的特征	112	11.2.2	隐藏常规处理	138
8.9.3	鸭子类型	113	11.2.3	替换部分算法	139
8.9.4	面向对象的例子	114	11.3	定义带块的方法	142
第 9 章 运算符	117		11.3.1	执行块	142
9.1	赋值运算符	117	11.3.2	传递块参数, 获取块的值	142
9.2	逻辑运算符的应用	118	11.3.3	控制块的执行	144
9.3	条件运算符	120	11.3.4	将块封装为对象	145
9.4	范围运算符	120	11.4	局部变量与块变量	147
9.5	运算符的优先级	121			

第 3 部分 Ruby 的类

第 12 章 数值类	150	12.4 Math 模块	155
12.1 数值类的构成	150	12.5 数值类型转换	156
12.2 数值的字面量	152	12.6 位运算	157
12.3 算数运算	153	12.7 随机数	159
除法	153	12.8 计数	160

12.9 近似值误差	161	14.3 字符串的索引	194
第 13 章 数组类	164	14.4 字符串的连接	194
13.1 复习数组	165	14.5 字符串的比较	195
13.2 数组的创建方法	165	字符串的大小比较	196
13.2.1 使用 Array.new	165	14.6 字符串的分割	198
13.2.2 使用 %w 与 %i	166	14.7 换行符的使用方法	198
13.2.3 使用 to_a 方法	166	14.8 字符串的检索与置换	199
13.2.4 使用字符串的 split 方法	166	14.8.1 字符串的检索	199
13.3 索引的使用方法	167	14.8.2 字符串的置换	200
13.3.1 获取元素	167	14.9 字符串与数组的共同方法	200
13.3.2 元素赋值	169	14.9.1 与索引操作相关的方法	200
13.3.3 插入元素	170	14.9.2 返回 Enumerator 对象的方法	201
13.3.4 通过多个索引创建数组	171	14.9.3 与连接、反转 (reverse) 相关的 方法	202
13.4 作为集合的数组	171	14.10 其他方法	203
" " 与 "+" 的不同点	173	14.11 日语字符编码的转换	204
13.5 作为列的数组	173	14.11.1 encode 方法	204
13.6 主要的数组方法	175	14.11.2 nkf 库	204
13.6.1 为数组添加元素	175	第 15 章 散列类	208
13.6.2 从数组中删除元素	177	15.1 复习散列	208
13.6.3 替换数组元素	179	15.2 散列的创建	209
13.7 数组与迭代器	180	15.2.1 使用 {}	209
13.8 处理数组中的元素	181	15.2.2 使用 Hash.new	209
13.8.1 使用循环与索引	181	15.3 值的获取与设定	210
13.8.2 使用 each 方法逐个获取元素	181	15.3.1 一次性获取所有的键、值	211
13.8.3 使用具有破坏性的方法实现循环	182	15.3.2 散列的默认值	211
13.8.4 使用其他迭代器	182	15.4 查看指定对象是否为散列的键或值	212
13.8.5 创建专用的迭代器	182	15.5 查看散列的大小	213
13.9 数组的元素	183	15.6 删除键值	213
13.9.1 使用简单的矩阵	183	15.7 初始化散列	214
13.9.2 初始化时的注意事项	183	处理有两个键的散列	215
13.10 同时访问多个数组	185	15.8 应用示例: 计算单词数量	216
第 14 章 字符串类	188	第 16 章 正则表达式类	219
14.1 字符串的创建	189	16.1 关于正则表达式	219
14.1.1 使用 %Q 与 %q	190	16.1.1 正则表达式的写法与用法	219
14.1.2 使用 Here Document	190	16.1.2 正则表达式对象的创建方法	220
14.1.3 使用 sprintf 方法	191	16.2 正则表达式的模式与匹配	220
14.1.4 使用 ``	191		
14.2 获取字符串的长度	193		

16.2.1 匹配普通字符	221	18.3 文件与目录的属性	259
16.2.2 匹配行首与行尾	221	FileTest 模块	261
16.2.3 指定匹配字符的范围	222	18.4 文件名的操作	262
16.2.4 匹配任意字符	224	18.5 与操作文件相关的库	263
16.2.5 使用反斜杠的模式	225	18.5.1 find 库	263
16.2.6 重复	226	18.5.2 tempfile 库	264
16.2.7 最短匹配	228	18.5.3 fileutils 库	265
16.2.8 () 与重复	229	第 19 章 Encoding 类	268
16.2.9 选择	229	19.1 Ruby 的编码与字符串	268
16.3 使用 quote 方法的正则表达式	229	19.2 脚本编码与魔法注释	269
16.4 正则表达式的选项	230	19.3 Encoding 类	270
16.5 捕获	231	Encoding 类的方法	271
16.6 使用正则表达式的方法	232	19.4 正则表达式与编码	274
16.6.1 sub 方法与 gsub 方法	232	19.5 IO 类与编码	274
16.6.2 scan 方法	233	19.5.1 外部编码与内部编码	274
16.7 正则表达式的例子	234	19.5.2 编码的设定	275
第 17 章 IO 类	237	19.5.3 编码的作用	276
17.1 输入 / 输出的种类	237	第 20 章 Time 类与 Date 类	280
17.1.1 标准输入 / 输出	237	20.1 Time 类与 Date 类	280
17.1.2 文件输入 / 输出	239	20.2 时间的获取	281
17.2 基本的输入 / 输出操作	240	20.3 时间的计算	282
17.2.1 输入操作	240	20.4 时间的格式	282
17.2.2 输出操作	243	20.5 本地时间	284
17.3 文件指针	244	20.6 从字符串中获取时间	284
17.4 二进制模式与文本模式	245	20.7 日期的获取	285
17.5 缓冲	246	20.8 日期的运算	286
17.6 与命令进行交互	248	20.9 日期的格式	286
17.7 open-uri 库	249	20.10 从字符串中获取日期	287
17.8 stringio 库	250	第 21 章 Proc 类	288
第 18 章 File 类与 Dir 类	252	21.1 Proc 类是什么	288
18.1 File 类	252	21.1.1 lambda	289
18.1.1 变更文件名	253	21.1.2 通过 Proc 参数接收块	291
18.1.2 复制文件	253	21.1.3 to_proc 方法	292
18.1.3 删除文件	254	21.2 Proc 的特征	292
18.2 目录的操作	254	21.3 Proc 类的实例方法	293
18.2.1 目录内容的读取	255		
18.2.2 目录的创建与删除	258		

第 4 部分 动手制作工具

第 22 章 文本处理 298

22.1 准备文本	298
22.1.1 下载文件	298
22.1.2 获取正文	299
22.1.3 删除标签	300
22.2 扩展 simple_grep.rb : 显示次数	302
计算匹配行	302
22.3 扩展 simple_grep.rb : 显示匹配的部分	304
22.3.1 突出匹配到的位置	304
22.3.2 显示前后各 10 个字符	305

22.3.3 让前后的字符数可变更	306
-------------------------	-----

第 23 章 检索邮政编码 308

23.1 获取邮政编码	308
23.2 检索邮政编码	309
23.3 sqlite3 库	309
23.4 插入数据	311
23.5 检索数据	313
23.6 总结	314

附录

附录 A Ruby 运行环境的构建 316

A.1 Ruby 的安装	316
A.2 在 Windows 下安装	316
A.2.1 开始安装	317
A.2.2 同意软件使用许可协议	317
A.2.3 确认安装路径以及选项	317
A.2.4 安装进度	318
A.2.5 安装完成	318
A.2.6 启动控制台	319
A.3 在 Mac OS X 下安装	320
通过 MacPorts 安装	321
A.4 在 Unix 下安装	322
A.4.1 从源代码编译	322
A.4.2 使用二进制软件包	323
A.4.3 使用 Ruby 软件包管理工具	323
A.5 编辑器与 IDE	323
没有好用的编辑器就不能编写 Ruby 程序吗	324

附录 B Ruby 参考集 325

B.1 RubyGems	325
--------------------	-----

gem 命令	325
--------------	-----

B.2 Ruby 参考手册 327

B.2.1 Web 上的资源	327
B.2.2 ri 命令	328
B.2.3 阅读参考手册的技巧	328

B.3 命令行选项 329

B.4 预定义变量、常量 330

B.4.1 预定义变量	330
B.4.2 预定义常量	331
B.4.3 伪变量	332
B.4.4 环境变量	332

B.5 错误信息 332

B.5.1 syntax error	333
B.5.2 NameError/NoMethodError	333
B.5.3 ArgumentError	333
B.5.4 TypeError	334
B.5.5 LoadError	334
B.5.6 [BUG]	334

后记 335

谢辞 336



第1部分

Ruby 初体验

让我们先从简单的程序入手，从整体上了解 Ruby 的基本概念，并对如何使用 Ruby 编写程序有个初步印象。