

- 《Ruby Programming——向Ruby之父学程序设计》姊妹篇
- 全方位展示Ruby程序设计技巧
- 活用Ruby的掌中宝典
- 兼顾Windows与UNIX，内容更全面

程序设计

268技

第2版

Rubyレシピブック 第2版268の技

【日】青木峰郎 后藤裕藏 高桥征义 著

【日】松本行弘 审校 博硕文化 译



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

程序设计

268 技

第2版

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Ruby 是为了让程序设计更快乐而开发的程序语言。Ruby 具有“彻底面向对象”、“丰富的程序库”、“亲和力高的直观语法形式”等多项特征，但这些特征并不是 Ruby 的目的，只能说是为了让程序设计更有趣而开发的手段罢了。

Ruby 处在当前主流的位置上，并非“行家才知道，专业人员为导向的程序语言”。在学习 Ruby 的人群当中，应该有很多人是看了参考手册后，不知道要用 Ruby 来做什么。本书并非以功能项目进行分类，而是以“想要做的事”作为主轴来进行分类，对于读者的学习绝对大有裨益。本书是在活用 Ruby 时不可离手的一本书，它能够帮助大家在程序设计的道路上获得更多的快乐。

希望本书的读者，都能从使用 Ruby 的过程中，找到属于自己的“兴趣”与“乐趣”。——若您可以这样灵活运用，那将是本书作者无上的荣幸了。

Ruby Recipe Book, Second Edition, 268 no Waza

Copyright© 2007 Minerou Aoki, Yuuzou Gotou, Masayoshi Takahashi

Copyright© Yukihito Matsumoto

Chinese translation rights in simplified characters arranged with Softbank Creative Corp., Tokyo through Japan UNI Agency, Inc., Tokyo01-2009-3249

本书中文简体版专有出版权由 Softbank Creative 公司授予电子工业出版社，专有出版权受法律保护。

版权贸易合同登记号 图字：01-2009-3249

图书在版编目（CIP）数据

Ruby 程序设计 268 技：第 2 版 / （日）青木峰郎，（日）后藤裕藏，（日）高桥征义著；博硕文化译。

北京：电子工业出版社，2009.7

ISBN 978-7-121-08812-4

I. R… II. ①青… ②后… ③高… ④博… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字（2009）第 073266 号

责任编辑：江 立

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：26.5 字数：560 千字

印 次：2009 年 7 月第 1 次印刷

定 价：59.00 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 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

第二版审校者序

由 2007 年初往前回顾，我认为《Ruby 程序设计》初版发行的 2004 年，对于 Ruby 来说是正值飞跃发展之际。同样是 2004 年所发表的 Web Application Framework——Ruby on Rails、“15 分钟建立一个 Blog 系统”的视频短片与“具有 Java 10 倍的生产力”的宣传口号，更是瞬时让 Ruby 在业界中备受瞩目。

我想也许是时间点恰恰好吧。Ruby 真的很幸运。这个时代越来越要求单一程序设计师其个人的生产性，另一方面计算机运算能力有着飞跃性进步，可以容许“浪费”计算能力。在 21 世纪的今日，像上面这样的动态程序语言一直都是众人所需，而在满足这些需求的程序语言中，Ruby 凭着 Ruby on Rails 的话题性，早一步到达这样的位置。此外，在 Mac OS X 上不仅仅将 Ruby 及 Ruby on Rails 当作标准的内建程序，也同时附加了控制 OS 的函数库，大有取得继 Objective-C 之后的开发程序语言地位的态势。

在现在 Ruby 与 Ruby on Rails 的周边环境，存在着“15 分钟建立一个简单 Blog 系统，有什么意义”、“真的有 Java 10 倍的生产力吗？！”等想法单纯的问题。而这些问题实际上无论是真是假都好，重要的是“Ruby 与 Ruby on Rails 就存在于当前，而且帮助程序设计师变得更快乐”这个事实。远在 Ruby 变得有名气之前开始，它就已经是一个对程序设计师很有用的工具了，而这一点从今以后我想应该也不会改变吧。

现在，Ruby 并非“行家才知道，专业人员为导向的程序语言”，而是处在当前主流的位置上。在学习 Ruby 的人群当中，应该有很多人是看了参考手册后，不知道要用 Ruby 来做什么。本书并非以功能项目进行分类，而是以“想要做的事”作为主轴来进行分类，对于各位的学习绝对可以帮上不少忙。我个人认为，本书是在大家活用 Ruby 时不可离手的一本书。若 Ruby 与本书能够帮助大家在程序设计的道路上获得更多的快乐，我会感到十分欣慰。

2007 年 1 月
于前往东京的班机上
松本行弘

初版审校者序

在设计一种新的程序语言时，会有三件难事。第一是设计一个“好的”程序语言，再来就是一本入门书，最后就是如何使用这个语言方法的 **Recipe** 手册（或者 **Cookbook**）。

对于 **Ruby** 来说，幸运的是第一件事已经解决了，但要克服接下来的事情就真的很不简单。入门书及 **Cookbook** 不论谁都希望能够出书，我们也企划了很多次，但始终“这本书”都没有出现。比起创造语言本身，我真的觉得执笔写出一本好书更加困难。

但是随着《**Ruby Programming——向 Ruby 之父学程序设计**》出版，第二项困难也被解决了，剩下的就是 **Cookbook**。接着，《**Ruby Programming——向 Ruby 之父学程序设计**》的作者们得到了强力的援军——《**Ruby 原始码完全解说**》的作者加入，终于完成了大家引颈期盼的《**Ruby 程序设计**》。真的是可喜可贺！

这本书最重要的就是帮助那些跨过 **Ruby** 门槛的人，在解决实际问题上收集各式各样的诀窍方法。虽然本书分析了在使用 **Ruby** 过程中各式各样的情况，但绝不会钻入难以了解的解释之中，而是细心又详细地解说到程序的每个小角落。虽然本书刻意地回避了 **Ruby** 所有功能中部分高度复杂以及难解的部分，但熟练本书中的内容之后，我想也是可以快速地凌驾 **Ruby** 的中级者程度吧。在此之后，就应该可以以自身的能力向更高等级前进。

我由衷希望本书能够成为各位在 **Ruby** 程序设计上的良友。

Happy Hacking.

松本行弘

写在前面

写了一个开源的程序并意外地被广泛使用这种事并不少。因工作关系写程序的不说，因自己的兴趣开始写程序，而刚好有机会被自己以外的人来使用。甚至，也常听到将程序发布到像 Web 一样许多人目光所及之处后，结果程序被远超过想象多的人来使用。

在刚开始写程序的时候，使用那个程序的用户，恐怕应该只有自己本身吧。起初因为学习等目的而开始写程序，在测试及确认可运作之后，就不会再被使用了。

像这样一个人来撰写程序，一个人使用，然后又丢掉的程序，使用怎样的撰写方式或许都无所谓。只要能够简单地写出来，能像预期的来运作就足够了。如果以料理来讲的话，一想到只有自己吃，就算食材已经超过保存期限，只要没有奇怪的味道就好了，或者是因为用盘子来装菜的话，要洗的盘子就变多，所以干脆吃炸面包就好了等，这样就会在不知不觉中越来越草率。就会想到“反正只跟我自己有关系，又不会影响到其他人”。

不过，一旦这些东西与其他人有交互时，就完全不同了。如果被某人招待的话，就会觉得不使用敬语是不行的。说到写程序也是一样的，用完就丢的一行指令与发布公开的脚本，写法一定会有所不同。

再者，写程序与做料理不同，会有许多人来进行，在未见过也未知的对方作品上来进行修改也不在少数。特别是以脚本语言写的程序，因为大多是以原始码的形式来流通，所以不论是谁都可以轻易地取得。因此，不单只是要让使用程序的人觉得好用，也要让写程序的人觉得好用，这样的程序才是我们所期待的。

总而言之，对我来说，追求的不只是写程序的快乐，而是要能够对写程序的人待若上宾。

当然，要达到这样的期望，不能只有想法，多样的程序语言知识也是必要的。这些知识包括了在某种程序语言中共同的知识，也包括只存在某个特定程序语言的特性。另外，也包括在许多情况下有用的技巧，以及在特定对象领域中不可欠缺的技巧。必须要熟悉各式各样的知识，然后加以使用。

再者，有时在待人的态度方法上也要多费心思。例如重新检阅算法、重新设计类别后进行测试、专注在方法名称及变量名称上等。这些工作大家或许都觉得很麻烦吧？不过，或许也正因为你花了这些工夫在程序上，而使得用户在使用时的愉悦更上一层楼说不定。这是非常幸福的一件事。然后，一天天地积累这些工作，不正是成就你写出更好程序的基石吗？

* * *

在 Perl 等程序语言中，有一种称为“食谱 (Cookbook)”的书。我们之前一直以写出这类书的 Ruby 版为目标开始执笔动工。不过或许实际上写出来的书其倾向会有点不同。把在 Perl 中的那类书当作教导读者做出满汉全席的话，我们写出来的这本书可能就是为普通人的基本家常菜而已，在程度上多少有些差异。不过，我们的目标是让这本书成为不输给那类书，而对读者能有很大帮助的一本书。

我想在这本书中你一定可以找到一些你所不知道的菜谱。而如果这些菜谱可以帮助你写出一个善待用户的程序的话，身为作者的我们就得到最大的喜悦了。

如果这本书能迎合您的口味的話，我们真的感到非常荣幸。

作者群

初版致谢

紧接着在《Ruby Programming——向 Ruby 之父学程序设计》之后，本书也得到许多人的帮助而能顺利到了出版的阶段。

最要感谢的人就是 SoftBank 出版社第一书籍编辑部的杉山先生。如果没有他坚强的耐心以及长时间跟进的话，本书绝对无法到达出版阶段。真的是非常感谢。

接下来要感谢的是 Ruby 的亲生父亲、给予我们审校帮助的松本行弘先生，以及帮我们审阅的 Nakamura、Nahi、Yamda Akira、Watanaba Hirofumi，以及安藤叶子小姐、沓名亮典先生、后藤谦太郎先生、西山和广先生、一步先生以及其他我们在本书中所使用函数库的开发者等，还有在 Ruby 的各 Mailing List 及 IRC 中提供各种有用以及有趣信息的社群朋友，还有 ruby-lang.org 各项服务的管理者，以及开发及维护 Ruby 的相关人员们，我们在此由衷向你们表示感谢。真的非常感谢你们的帮助。

还有，本书原本只预定由《Ruby Programming——向 Ruby 之父学程序设计》的两位作者后藤以及高桥先生来执笔，不过在苦战的结尾，我们向青木先生寻求支持，而形成三人的团队。在最终的原稿执笔上，青木先生的贡献是最大的。虽然是作者群中的一人，也不能说得上是致谢，但在此还是要记下本书能顺利发行的经过。

关于本书

■ 本书的对象及其操作环境

本书主要的对象为

- 对于 Ruby 的概要已经了解的读者
- 虽然不懂 Ruby，但已经学过其他程序语言的读者

而对于那些第一次接触程序设计以及 Ruby 的人来说，我们并不推荐您阅读此书。不过我们推荐您阅读本书的姊妹篇——《Ruby Programming——向 Ruby 之父学程序设计》。

另外，本书的操作环境为 Ruby 1.6.7 版至 1.8.5 版。其中我们会特别将重点放在 1.8.5 版上。

部分与版本有关会造成不同结果的程序代码，我们会在该项技巧的“说明”部分补充有关版本间兼容性的问题。

在操作系统环境部分，工作平台可以是 Windows 及各种 UNIX。特别与工作环境有关的脚本，请各位读者确认是否为以下的操作系统。

- Windows XP SP2 / mswin32 版 Ruby (i686)
- Windows XP SP2 / Cygwin 版 Ruby (i686)
- Windows 2000 SP4 / mswin32 版 Ruby (i686)
- Windows 2000 SP4 / Cygwin 版 Ruby (i686)
- Debian 3.1 (Linux 2.6 / AMD64)
- Debian 3.1 (Linux 2.6 / i686)
- Solaris 9 (SPARC 32bit)
- Tru64UNIX 5.1B (Alpha)

各个技巧 (Recipe)，可由 、 的标示来知道该技巧是否可在 Windows 或 UNIX 环境下使用。

■ 给使用 Windows 的使用者

可以在 Windows 下执行的 Ruby 程序有许多种，其中以 Cygwin 版与 UNIX 的操作最为相近，所以这个版本是包含在  标示中，而不是  标示。

■ 正文中的表示方式

● 程序的输出

书中程序的输出内容会以“输出方法的侧边，或者在次行加上 `#=>`，然后其右侧为输出结果”的方式来表示，如：

```
# 以 p 方法来显示 "Hello, Ruby."  
p "Hello, Ruby."      #=> "Hello, Ruby."
```

这种方式在 Ruby 的使用手册中经常使用。

● 实例方法与类别方法

本书在方法的表示方式上也是采用在 Ruby 使用手册中常用的表示方式。

实例方法使用“类别名称#方法名称”来表示。例如 `String` 类别的实例方法 `length`，本书中会以 `String#length` 来表示。这种表示方式是在文件或者说明中使用的，若在程序中使用的话，则会发生错误，请读者要特别注意。

类别方法在本书中会以“类别名称.方法名称”来表示。例如 `Array` 类别的类别方法 `new`，在本书中会以 `Array.new` 来表示。这种表示方法与实际在程序中的表示方法是相同的。

目



第 1 章

Ruby 的语法

001	常值 (literal)	2
002	运算符.....	6
003	变量与常数.....	9
004	代入.....	11
005	程序语句的分段.....	12
006	备注.....	13
007	真假值与条件判断.....	14
008	循环.....	17
009	调用方法.....	19
010	区块 (block)	20
011	方法 (Method) 定义	21
012	限制方法的调用.....	22
013	类别 (Class) 定义	23
014	模块 (Module) 定义	24
015	特殊方法.....	25
016	例外处理.....	26
017	alias 与 undef	27
018	保留字.....	28

第 2 章

字符串

019	选择适当的字符串常值.....	30
020	设置默认的文字编码.....	32
021	变换文字编码.....	34
022	推测文字编码.....	37
023	复制字符串.....	39
024	重复字符串.....	40

025	取得字符串长度.....	41
026	查找字符串.....	42
027	查询正则表达式是否匹配字符串的开头/结尾	45
028	匹配单字.....	46
029	以特定的文字编码来进行正则表达式匹配	48
030	在正则表达式 “.” 中匹配 \n	50
031	从上次匹配的部分开始继续匹配.....	51
032	计算特定的文字、字符串出现的次数	53
033	计算字符的出现频率.....	55
034	比较字符串.....	56
035	对空白进行匹配.....	57
036	判断字符串是否为空行.....	58
037	判断是否为表示数值的字符串.....	59
038	将字符串变换为数值.....	60
039	判断字符为大写或小写.....	62
040	将字母变换为大写/小写	63
041	取出字符串的一部分.....	65
042	取出正则表达式匹配的部分.....	67
043	取出正则表达式匹配部分的前后字符串	69
044	查询包含特定文字部分的长度.....	70
045	将字符串分割为段落.....	71
046	将字符串分割为行.....	73
047	将字符串分割为单字.....	74
048	以正则表达式来分割字符串.....	76
049	处理字符串的个别文字.....	78
050	处理字符串的单一字节.....	79
051	链接字符串.....	80
052	置换字符串的一部分内容.....	81
053	跳脱字符/除去跳脱字符	83
054	除去字符串的一部分.....	85
055	取出行尾的换行字符.....	87
056	除去字符串开头/结尾的空白	89
057	变换 Tab 与半角空格符.....	91
058	变更缩排.....	92

059	将 Here Document 缩排后编写	94
060	在字符串结尾加上字符串	95
061	对字符串的一部分进行破坏性删除	97
062	在字符串中间破坏性地插入字符串	99
063	字符与字符编码相互变换	101
064	将字符串依 n 个文字来分割	103
065	将字符串切出最大 n 字节的部分字符串	104
066	将字符串切出最大 n 个文字的部分字符串	106
067	将字符串切出最大 n 位的部分字符串	107
068	以特定文字来隐藏部分字符串	108
069	建立信息摘要 (Message Digest)	109
070	判断字符串是否可作为正确的正则表达式	111
071	跳脱正则表达式的 Meta 文字	112
072	判断字符串是否同时与多个正则表达式匹配	113
073	分割正则表达式来记述	114
074	正则表达式集	116
075	将字符串整形为各段落	118
076	解析各种文字格式	119
077	处理以 “,” 来分隔的数据 (CSV)	122
078	解析 XML	124
079	解析 YAML	127
080	处理 HTML	129
081	自动补完单字	132
082	判断是否为有效的 E-mail 地址	134

第 3 章 数组 (Array) 与哈希 (Hash)

083	建立数组	136
084	复制数组	138
085	建立多维数组	139
086	依序处理数组及哈希的所有元素	140
087	并行处理多个数组	141
088	多维数组的循环	143
089	取得数组的元素	144
090	改变数组的元素	146

091	删除数组的元素.....	147
092	在数组的开头/结尾加入元素	148
093	取出并除去数组开头/结尾的元素	149
094	置换数组的元素.....	150
095	随机取得数组中的元素.....	151
096	随机排列数组中的元素.....	152
097	查询数组的元素个数.....	153
098	查询数组中元素的出现次数.....	154
099	从数组中一次取出多个元素.....	155
100	取得数组中索引与元素的配对组.....	156
101	将数组中元素填入相同的值.....	157
102	由数组中取出并除去重复的元素.....	158
103	取得数组中满足条件的元素.....	159
104	查找数组的元素.....	160
105	比较两个数组.....	161
106	找出两个数组中相同的元素.....	162
107	取出两个数组间的差.....	163
108	连接数组.....	164
109	将数组中元素的顺序反转.....	165
110	将一定范围的整数作为数组的元素	166
111	将数组作为（以“,”）连接的字符串	167
112	二元查找法.....	168
113	建立哈希的键值类别.....	169
114	在哈希中加入元素.....	170
115	查询哈希内键值是否存在.....	171
116	删除哈希的元素.....	172
117	显示哈希的内容.....	173
118	将哈希元素以插入顺序取出.....	174
119	建立单一键可对应多值的哈希.....	176
120	交替哈希的键与值.....	177
121	排序数组及哈希.....	178
122	合并哈希.....	180
123	找出两个哈希中都有的键/只有一个哈希才有的键	181
124	设置默认值.....	182

125	使用巨大的离散数组.....	183
126	建立具有 Collection 特性的类别.....	184
127	使用堆栈与队列.....	185

第 4 章 文件与目录

128	将文件名分割为其组成元素.....	188
129	从相对路径求出绝对路径.....	189
130	判断路径名是否吻合样式 (Pattern)	190
131	取得目录与文件一览.....	192
132	取得目前所在的目录名称.....	194
133	变更当前目录.....	195
134	求得执行脚本所处的路径.....	196
135	更改文件名.....	197
136	复制文件.....	198
137	删除文件.....	200
138	判断文件是否可读取、可写入.....	201
139	判断文件是否存在/判断其种类	202
140	取得文件属性.....	203
141	变更文件属性.....	205
142	建立链接.....	207
143	判断两个文件是否相同.....	209
144	建立目录.....	211
145	对目录内的所有文件进行递归处理	212
146	递归复制目录.....	213
147	删除目录.....	214
148	事先确认文件操作.....	215
149	新增名称不重复的文件.....	216

第 5 章 输入/输出

150	打开文件.....	219
151	以行为单位来处理文件.....	220
152	读取文件内容.....	222
153	建立新的文件.....	224

154	在文件末尾加入数据.....	226
155	以单行指令 (One-liner) 更新文件	227
156	使输出操作与文件更新同步.....	228
157	计算文件的行数.....	229
158	将文件由文件末尾开始处理至行单位中	230
159	读取正在写入的文件.....	232
160	建立暂存文件.....	233
161	锁定文件.....	234
162	将文件转为十六进制.....	235
163	在文件中插入一行.....	236
164	以二进制模式写入文件.....	238
165	处理固定长度的记录.....	239
166	切割文件大小.....	241
167	删除文件开头的 n 行.....	242
168	删除文件末尾的 n 行.....	243
169	将标准输入/输出接替至文件	245
170	复制 I/O 对象	246
171	以 printf 来格式化输出.....	248

第 6 章 数值

172	格式化数值.....	251
173	位运算.....	253
174	在二进制、八进制、十进制、十六进制间相互转换	254
175	将数值每三位以 “,” 区隔开来.....	255
176	指定次数并重复处理.....	256
177	求解最大值、最小值.....	257
178	求解数值的总和.....	258
179	求绝对值.....	259
180	求除法的余数.....	260
181	求解指数.....	262
182	求最大公因数及最小公倍数.....	263
183	求解平方根.....	264
184	计算三角函数.....	265
185	将直角坐标系转换为极坐标系.....	266

186	进行对数计算.....	267
187	对浮点数进行四舍五入等处理.....	268
188	比较浮点数.....	270
189	取得随机数.....	271
190	使用矩阵.....	272
191	使用复数.....	274
192	使用有理数.....	275

第 7 章 日期与时间

193	查询现在的日期与时间.....	277
194	由日期时间中取出月份、星期几等.....	279
195	日期时间与 Epoch 秒相互转换.....	280
196	求两个日期时间的差.....	282
197	格式化日期时间.....	283
198	将字符串转换为日期时间.....	285
199	求出由 1 月 1 日起到目前所经过的天数.....	286
200	判断闰年.....	287
201	求出一个月最后一天的日期.....	288
202	判断是否为假日.....	289
203	求出“本月第三个星期一”的日期.....	290

第 8 章 环境相关

204	将 Ruby 脚本做成一般指令.....	292
205	暂时性地在做成指令的 Ruby 脚本上加入选项后执行.....	295
206	取得命令行的参数.....	296
207	解析命令行选项.....	298
208	存取环境变量.....	301
209	让脚本作为指令或者函数库都可以执行.....	302
210	控制 Ruby 的函数库加载器 (Library Loader).....	303
211	对 Ruby 脚本进行除错.....	304
212	取得用户及群组相关信息.....	307
213	切换使用者 ID、群组 ID.....	310
214	存取 Windows 登录文件 (Registry).....	312
215	调用系统函数库的函数.....	313