

于天恩 编著 ■

Ruby

几门权威经典

```
module My
  NA="China"
  def My.set_name(name)
    @name=name
  end
  def My.get_name
    return @name
  end
  def My.set_age(age)
    @age=age
  end
end
```

```
My.set_name("tianen")
p My.get_name
p My::NA
```

```
module Your
  NA="AC"
  attr: name
  def Your.set_name(name)
    My.set_name(name)
  end
  def Your.get_name
```



北京航空航天大学出版社

Ruby 入门权威经典

于天恩 编著

北京航空航天大学出版社

内 容 简 介

本书基于 Ruby 的 186-25 版本进行讲解。全书共包括 8 章,包含与 Ruby 编程相关的一切核心基础知识以及使用 Ruby 开发常规 Web 应用程序的方法,可以分成三个部分。第一部分(第 1 章):Ruby 简介和环境配置。介绍了 Ruby 语言及 Ruby 框架的特点以及配置开发平台的方法。第二部分(第 2~4 章):Ruby 的基础知识。介绍了 Ruby 语言的基本结构、流程控制、数据类型、模块和线程等基础知识。第三部分(第 5~8 章):Ruby 的高级知识。介绍了 Ruby 的文件和目录操作以及数据库操作方法。最后提供了一些案例,用以实践 Ruby 的 Web 开发。

本书适用于对 Ruby 开发感兴趣的院校学生以及专业工程师。

图书在版编目(CIP)数据

Ruby 入门权威经典/于天恩编著. —北京:北京航空航天大学出版社,2009.4

ISBN 978-7-81124-576-9

I. R… II. 于… III. 程序语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 199245 号

Ruby 入门权威经典

于天恩 编著

责任编辑 罗晓莉

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:bhpressell@263.net

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1 092 1/16 印张:16.75 字数:429 千字

2009 年 4 月第 1 版 2009 年 4 月第 1 次印刷 印数:3 000 册

ISBN 978-7-81124-576-9 定价:28.80 元

前 言

说说 Ruby

Ruby 是一种语言,其单词含义为“红宝石”,许多女孩都用 Ruby 作为自己的名字。

既然是语言,当然是用来编程的。使用 Ruby 进行编程的效率很高,严格地讲是特别高。它是专门用来进行高效编程的,学习起来也很轻松。

Rails 是一种技术框架,其单词含义为“铁轨”,于是“Ruby On Rails”的含义就是铺满红宝石的铁轨。Rails 是 B/S 结构的编程框架,基于 Ruby 语言。这类似于 ASP 基于 VBScript 和 JSP(Struts)基于 Java。

单独学习 Ruby 是可以的,你可以使用 Ruby 去编写各种程序。不过,通常人们不会这么做,对我而言,最好的万能语言是 Java,用 JBuilder 开发 Java 程序是非常舒服的。

如果你也不打算使用 Ruby 作为万能语言的话,那么,最好使用它来做 Web 开发,也就是使用 Ruby On Rails 框架,这是 Ruby 最好的应用。

要学习 Ruby On Rails,首先要学习 Ruby。个人认为使用 Ruby On Rails 做 Web 开发的效率是非常高的,比 JSP,ASP,PHP 都要高,和 ASP.NET 2.0 也不相上下。尤其,Ruby On Rails 是开源的,免费的,因此,做 Web 开发,它是优于 ASP.NET 2.0 的选择。但,它是解释执行的,而不是编译执行的,这导致它和 JSP,ASP.NET 相比,在执行效率上有些差距。

虽然我没法证明 Ruby On Rails 是最佳的 Web 开发技术,但它在许多时候是我做开发的首选。推己及人,建议你使用,不是忽悠你。

写作动机

写书以前,我问了一个问题:外国权威书写得不好吗?国人翻译的不好吗?为什么还要我写?

Ruby 和 Rails 的书全世界有两本好的,这就够了,那些烂书我们可以不去看它。这是我原来的想法。

但论坛上许多人提出了重复的问题,他们在学习 Ruby 时捉襟见肘,不得要领。我向一个不懂 Ruby 的计算机博士推荐 Ruby 语言,他居然花了好长的时间都没有学会。这使我感到惊讶,在详细地了解了他的难处之后,我才明白国人遇到的问题。我开始觉得有必要写一本适合中国人学习的 Ruby 教材。

国人学习 Ruby,许多是认为这技术新鲜,有前途,能赚钱。Ruby 是一门总结性的优化型的语言,如果学习者具有 Java,C# 等语言的开发经验的话,要学习这门语言是非常轻松的。但多数人没有什么编程经验就直接学习 Ruby,这样一来,那些国外的权威著作就不太适合了。国人需要的是直接从零起点来讲解 Ruby 的书籍。

正因为此,我放下宝贵的休息时间,开始了此书创作。

写作过程

作为一本基础教程,一本入门教程,写作的关注点是知识的次序,而不是知识的全面性和系统性。所以,我在这本书中充分考虑到初学者的实际情况,尽可能合理地编排次序,在这方面花去了不少时间。初稿完成后印刷给别人试用,收到了良好效果。之后联系出版社出版。然后,这本书就到了你——读者的手中。

本书特点

写书,应以教育为目的。此书内容以实践为主,并不深挖理论。教育是目的,写书是方法,方法服从于目的。书,不见得要写得多深多透。书的好坏要由书所面向的读者群来评价。我写书的原则是:实用。

本书包含了必要的理论,但以实践为主。所讲的理论都不是纸上谈兵,都是可以立即付诸实践进行工程应用的。代码可以直接拿出来用,只是不要忘了输入信息验证等基本的安全检查。

书中含有大量的案例,讲解由浅入深。浅,并不从“什么是程序设计”开始,所以读者需要具备一些编程的基础知识才能看懂。深,并没有深到“只可意会,无法言传”的地步,所以读者不需要担心无法看懂。

当然,我在书中的确阐述了一些思想,这是因为:我以为教育不单是要传知识,更重要地是要传思想。李阳的《疯狂英语》、胡敏的《新魅力英语》都是在给人以信念、生活的动力。一本书,如果能够启迪读者的思维,使读者找到自己人生的方向,那便是功德无量了。

本书面向初学者,专门为初学者编写。初学者并不仅仅意味着“不懂”和“不会”,还意味着“没有受到错误思想的污染”。

写一本大全,或者心得,之后或许能得到“厚重”,“精深”的美誉。但就当前的现实情况而言,写基础教程比写大全更有意义。基础教程面向的读者多,看到大量的读者通过我的书学到知识,改善自己乃至父母的生活,我会很开心。中国需要 Ruby On Rails 的基础教材,很多人要学习这门技术却不知道从何处入手,他们就是我写书的读者群。

这本书就是要让你知道:原来 Ruby 如此简单!

本书保持了我一贯的风格,以循序渐进为写作原则,先介绍概念然后再讲应用,应用程序由简入繁。这看起来像废话,多简单啊!每个人都这样啊!说实话,这的确是废话,但就是这么一件简单到掉渣的事情,却没几个人能做好。有人写 Rails 基础教材,先写一堆 Rails 心得,然后讲 Ruby。也有人写 Java Web 开发教材,先讲 JSP 后讲 Servlet。我猜这书是写给他们自己看的。所以,世界上的书不少,但有资格作为基础教材的还真是凤毛麟角。

这本书的循序渐进,读者可以体会得到。写一本 Ruby 的基础教材不难,但要把循序渐进贯彻其中,就要用心编排了,这是难点。介绍运算符之前绝对不使用运算符,介绍变量之前,不使用变量。所有出现的知识都在前面有所铺垫。这是我希望做到的。

我这本书不是面向天才的,天才学 Ruby 哪里需要看书啊,只要浏览一下技术文档也就足

够了,最多几个小时就精通了。

我这本书面向的是你,需要技术,需要工作,需要赚钱,需要养家的你。衷心祝愿你成功!努力!奋斗!

本书内容

这本书基于 Ruby 的 186-25 版本来讲解。全书共包括 8 章,可以分成三个部分。你将从这里学习到和 Ruby 编程相关的一切核心基础知识以及使用 Ruby 开发常规 Web 应用程序的方法。

第一部分(第 1 章):Ruby 简介和环境配置。介绍了 Ruby 语言及 Ruby On Rails 框架的特点以及配置开发平台的方法。

第二部分(第 2~4 章):Ruby 的基础知识。介绍了 Ruby 语言的基本结构、流程控制、数据类型、模块和线程等基础知识。

第三部分(第 5~8 章):Ruby 的高级知识。介绍了 Ruby 的文件和目录操作以及数据库操作方法。最后提供了一些案例,用以实践 Ruby 的 Web 开发。

Ruby 包含的内容很多,操作系统编程、网络编程等应有尽有。手册里面写得很详细。但我写的是基础教程,所以不能面面俱到,写的是基础知识中的基础,目的是使读者入门,然后再向前走一步以进行常规应用开发。同时,恰恰由于这本书是基础中的基础,因此,在 Ruby 迅速发展并且不时发生巨大变化的现实情况下,本书将始终保持着生命力。

学完这本书,读者就可以掌握所有 Ruby 核心基础知识并应付常规开发,可以使用 eruby 解决常规的 Web 开发问题。读者还将具备看懂高级 Ruby 和 Rails 资料的能力。这便是本书最浅显的意义。

之后,如果读者学有余力,想要继续学习 Ruby 和 Rails,那么我推荐《Programming Ruby》和《Agile Web Development with Rails》。对这两本书,我曾认真地看了,很不错,都是很好的书,是同类中的翘楚。

谅解和支持

这本书是我用心写成的。

我喜欢读书,读过很多书,读好书对我来说是享受,它能减少我的思考时间,哪怕只有几分钟,也是相当值得的。做人做事得将心比心,我写的每一本书都是用心的,用心,尽力做好。这样就可以节约你们的时间。

我尽心尽力地写了这本书,从章节的安排到案例的编写,都是经过仔细揣摩的。我希望做到最好。然而,没有最好,只有更好。

我不能确定这里面没有错误,但我很希望经过多次的检查可以使里面的错误减到最少。即使有错,也应该是文字或者其他一些错误,而不会有涉及程序设计或算法的原则性错误。

我希望你说这是一本好书,但我更希望你告诉我这本书的缺点和不足。因为我知道这里一定有不足。我不能因为自己用了心,就认为自己做到了完美。问心有没有愧是一回事,做得

好不好是另一回事。前者,我说了算,后者,你们说了算。

如果你在本书中发现任何问题,我都很希望你能告诉我,以帮助我改良它。而且,你可能收到意想不到的神秘礼物,或许是我珍藏十几年的树叶书签,或许是我亲手叠的纸飞机,或许是我写的诗词……总之,有钱买不到!

我的邮箱:yutianen@163.com 将真诚地用来为读者答疑解惑,同时接受所有读者的反馈意见。需要说明的是,我工作之外的时间非常少,如果不能及时查收或回复你的邮件,希望谅解。我若看到了你的邮件,就一定会认真回复的。

希望这本书在你做 Ruby 应用开发时有帮助,那便是我的欣慰。

如果这是你学习 Ruby 的第一本教材,我赞赏你的眼光,你选中了一本用心写的书。

愿这本书带给你知识,带给你幸运,带给你真实的人生。

一些勉励

艺多不压身。

家财万贯不如艺在身。

学技术很有用,可以让心里踏实。只要技术活着,至少我们可以靠技术活着。技术死了,我们再去学新技术,从而维持自己活着。这是一种与时俱进。

只要肯钻研,没有学不通的技术。

人类若不灭亡,技术就无穷无尽。只要有兴趣,有动力,就可以不停地学,锲而不舍,最终把所有的技术都学会。

我不建议把所有的心思都用在技术上。我更加不认为技术的进步就是人类的进步,人类的社会,始不可知,终无可料。不要执着于任何事情,别想不开。

回到技术,诚然,想要掌握主流的所有软件开发理论和技术并不是什么难事,确切地说这非常简单。但,如果你为了学技术而学技术,而且并非天资聪颖轻松学会,而是花了大力气去苦学技术,那就太不值得了。看看你失去了多少和亲人朋友在一起的时间,看看你失去了多少登山野游的时间,看看你在感情和健康方面的损失,想明白这个事情,不要舍本逐末。

生而知之者,上也;学而知之者,次也;困而学之者,又其次也。

“人事有可陷者,亦有不可陷者”,陷于技术,不值得。

悟性,因人而异,不可强求,欲望,因人而异,不应执着。

学技术是简单的事情,做工程师是简单的事情,做科学家也是简单的事情,这些事都比做思想工作轻松得多。

但简单的事情未必好办,因为简单是相对的。不过有一点是肯定的,“世上无难事,只怕有心人”。即使我们无法确定自己可以做成每件事情,但至少我们可以对想做的事情用心,尽力,做到问心无愧。

诚然,人的天资很重要,非常重要,相当重要。但有几句话要提醒读者:“勤能补拙是良训,一分辛劳一分才”,“只要有恒心,铁杵磨成针”。天资是天给的,不是人定的。但在天资之外,你还有很多事情可以做,通过勤奋、坚持、刻苦,纵然超不过天才,你也定然能成为超越常人的

人才,这不是也很好吗?

幸福在哪里? 在精心的耕耘中,在艰苦的劳动里。

努力吧! 用晶莹的汗水去酿造成功,用辛勤的双手去编织绚丽的人生!

衷心感谢

石志国,他的《ASP 精解案例教程》是我编程的开端,提起编程,总会想起。想起 2004 的初冬在寒风凛冽的雨夜读书的事情,想起那时陪伴我的歌,不禁潸然泪下。

对哈工大天萌联合的一切成员表示感谢! 那些曾跟我在一起的朋友,我会记得你们为我泡的每一杯咖啡和茶。那些始终保持独立的朋友,我也祝愿你们会有更加辉煌的未来。我创立的天萌联合,我们的天萌联合,永远是哈工大最强、最自由的社团,你们这些天萌的元老的名字,将永远铭刻在我的心里。

本书原代码可以在北京航空航天大学出版社网站(<http://www.buaapress.com>)下载中心下载。

于天恩 中国·天人居
2008 年 11 月

目 录

第一部分 Ruby 简介和环境配置

第 1 章 配置 Ruby 运行环境	1
1.1 面向对象和 MVC 模式	1
1.1.1 面向对象	1
1.1.2 MVC 模式	1
1.2 Ruby 概述	2
1.2.1 Ruby 的由来	2
1.2.2 Ruby 的优点	2
1.3 Ruby 的安装	4
1.3.1 下载 Ruby	4
1.3.2 安装 Ruby	7
小 结	11
思考和练习	11

第二部分 Ruby 的基础知识

第 2 章 Ruby 语言基础	12
2.1 最基本的知识	12
2.1.1 基本输出	12
2.1.2 引号的用法	13
2.1.3 转义字符输出	14
2.1.4 连句和换行	15
2.1.5 连行符	17
2.1.6 注释符	18
2.1.7 局部变量	18
2.1.8 基本输入	19
2.1.9 数字和字符串连接	19
2.2 类	20
2.2.1 最简单的类	21
2.2.2 类的基本使用	21
2.2.3 继 承	22

2.2.4	单态方法	24
2.2.5	方法访问控制	25
2.2.6	属性读写控制	27
2.2.7	垃圾收集	31
2.2.8	异常处理	31
2.3	常量和变量	34
2.3.1	常量	35
2.3.2	全局变量	35
2.3.3	实例变量	37
2.3.4	局部变量	37
2.4	运算符	37
2.4.1	算术运算符	37
2.4.2	关系运算符	38
2.4.3	逻辑运算符	39
2.4.4	其他运算符	40
2.5	流程控制	40
2.5.1	顺序结构	40
2.5.2	选择结构	41
2.5.3	循环结构	45
2.6	块	48
2.6.1	块的概念	48
2.6.2	块的基本使用	49
2.6.3	带参数的块	50
2.7	迭代器	51
2.7.1	迭代的概念	51
2.7.2	编写迭代器	52
2.8	过程对象	54
2.8.1	创建过程对象	55
2.8.2	把过程对象当作参数	55
小 结		56
思考和练习		56

第3章 Ruby 的数据类型

57

3.1	数 字	57
3.1.1	数字的基本使用	57
3.1.2	数字的常用方法	59
3.1.3	数学计算方法	60

3.2 字符串	62
3.2.1 字符串的基本用法	62
3.2.2 字符串的常用方法	63
3.2.3 字符串方法总结	91
3.3 正则表达式	92
3.3.1 Ruby 正则表达式的基本用法	92
3.3.2 正则表达式在字符串函数中的使用	94
3.4 日期和时间	94
3.4.1 Time 对象	94
3.4.2 Date 和 DateTime 对象	98
3.5 散列表	99
3.5.1 散列表的构造	99
3.5.2 散列表的常用方法	100
3.6 区间	103
3.6.1 区间的概念	103
3.6.2 区间的使用	104
3.7 数组	106
3.7.1 构造数组	106
3.7.2 数组的主要方法	107
3.8 结构体	114
3.8.1 建立结构体	114
3.8.2 结构体的主要方法	115
3.9 数据类型转换	116
3.9.1 通用的转换方法	116
3.9.2 自定义转换方法	118
小 结	118
思考和练习	118

第4章 模块和线程 119

4.1 模 块	119
4.1.1 模块的概念	119
4.1.2 Mixin	122
4.1.3 Ruby 的命名约定	122
4.2 线 程	123
4.2.1 线程的概念	123
4.2.2 线程的同步	125
小 结	128
思考和练习	128

第三部分 Ruby 的高级知识

第 5 章 文件和目录 129

5.1 文件操作	129
5.1.1 文件操作的概念	129
5.1.2 文件的基本操作方法	129
5.1.3 文件操作标准方法	138
5.2 目录操作	146
5.2.1 目录操作的概念	146
5.2.2 目录操作的方法	146
小 结	153
思考和练习	153

第 6 章 Ruby 的数据库操作 154

6.1 Ruby 数据库访问的概念	154
6.1.1 数据库访问的方式	154
6.1.2 数据库访问的目的	154
6.2 访问 Access 数据库	154
6.2.1 配置环境	154
6.2.2 执行数据操纵语句	158
6.2.3 执行数据查询语句	163
6.3 访问 SQL Server 数据库	170
6.3.1 建立 ODBC 数据源	170
6.3.2 访问数据库	174
6.4 访问 MySQL 数据库	179
6.4.1 下载和安装 MySQL/Ruby 模块	179
6.4.2 使用 MySQL 模块进行数据库访问	181
6.4.3 安装 DBI	192
6.4.4 使用 DBI 访问 MySQL 数据库	197
小 结	201
思考和练习	201

第 7 章 桌面应用和 Web 开发 202

7.1 Ruby 的桌面开发	202
7.2 Ruby 的 Web 开发方法	202
7.2.1 CGI 类	202

目 录

7.2.2 eRuby 概述	204
7.2.3 eruby 的基本使用	207
7.2.4 文件包含	208
7.2.5 中文显示	209
7.2.6 参数的传递和接收	211
7.3 详解表单处理	214
7.3.1 表单的提交	214
7.3.2 表单的接收	214
7.4 文件操作	218
7.4.1 文件读取	218
7.4.2 文件写入	220
7.5 数据库操作	222
7.5.1 数据读取	222
7.5.2 分页显示数据	224
7.5.3 数据更新	229
7.6 session 的基本用法	233
7.7 Web 开发案例	234
7.7.1 留言本(基于文本文件)	234
7.7.2 聊天室(基于文本文件)	237
7.7.3 留言本(基于数据库)	242
7.7.4 聊天室(基于数据库)	245
小 结	251
思考和练习	251

第一部分 Ruby on Rails 简介和环境配置

第 1 章 配置运行环境

本章要点

本章首先介绍 Ruby 和 Rails 的历史和发展状况,以及编程特点,进而使读者从整体上了解这种编程技术。然后介绍配置 Ruby on Rails 运行环境的方法。

1.1 面向对象和 MVC 模式

在接触 Ruby on Rails 之前,我们先来看以下这两个概念。这是两个基本概念,也是和 Ruby on Rails 关系最紧密的概念。

1.1.1 面向对象

面向对象是一种程序设计方法,其基本思想是使用对象、类、继承、封装和消息等基本概念来进行程序设计。

它是从现实世界中客观存在的事物(即对象)出发来构造软件系统,并在系统构造中尽可能运用人类的自然思维方式,强调直接以事物为中心来认识问题、思考问题,并根据这些事物的本质特征,把它们抽象地表示为系统中的对象,作为系统的基本构成单位。这可以使系统直接地映射问题域,保持问题域中事物及其相互关系的本来面貌。

从程序设计的角度来看,面向对象的程序设计语言必须有描述对象及其相互之间关系的语言成分。这些程序设计语言可以归纳为以下几类:系统中一切皆为对象;对象是属性及其操作的封装体;对象可按其性质划分为类,对象成为类的实例;实例关系和继承关系是对象之间的静态关系;消息传递是对象之间动态联系的唯一形式,也是计算的唯一形式;方法是消息的序列。

因为市面上介绍面向对象理论的书非常多,所以本书不详细介绍这方面的内容。后面的讲解中将假设读者已经了解面向对象理论,所以不会对“类、对象、方法、继承”这些概念进行专门的论述。

Ruby 是纯粹的面向对象语言。

1.1.2 MVC 模式

MVC 模式即 Model - View - Controller 模式,中文翻译为模型-视图-控制器。MVC 应用程序总是由这 3 个部分组成。事件导致控制器改变模型或视图,或者同时改变两者。只要控制器改变了模型的数据,所有依赖的视图都会自动更新。类似地,只要控制器改变了视图,视图会从潜在的模型中获取数据来刷新自己。MVC 模式最早是 Smalltalk 语言研究团提出

的,应用于用户交互应用程序中。

这种模式具有清晰的逻辑划分,Controller 用来处理请求的事务,负责响应客户对业务逻辑的请求并根据用户的请求行为,决定将哪个 View 页面发送给客户。Model 则负责数据的处理。其结构如图 1-1 所示。

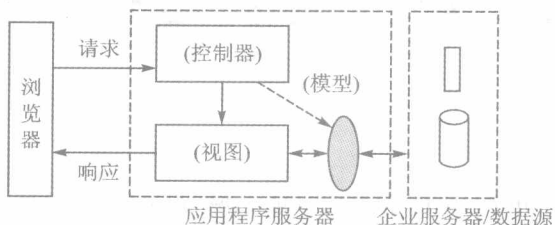


图 1-1 MVC 结构

MVC 可以使每项技术都发挥各自的长处,初始的请求由控制器来处理。控制器调用商业逻辑和数据处理代码,并创建模型来表示相应的结果。然后控制器确定哪个页面适合于表达这些特定的结果,并将请求转发到相应的页面(页面即为视图)。

MVC 模式能够有效地区分不同的开发者,避免彼此间的互相影响,充分发挥每个人的特长,在开发大型项目时表现出的优势尤其突出。

Rails 是完善的 MVC 模式的 Web 框架。

1.2 Ruby 概述

1.2.1 Ruby 的由来

Ruby 的作者是日本人,名为松本行弘。此人很喜欢编程,也精通很多语言,对 Perl 和 Python 有较深了解。为了编写一个最好的脚本语言,他吸取了 Perl 和 Python 等语言的精华(当然,在他看来是精华),写成了 Ruby。

Ruby 是 1995 年公开发布的。最开始的名字叫 Red Stone(红石头)。这个名字长,而且生硬,不太好听,所以改为 Ruby(红宝石)。前者是男人名字,透着刚强,后者是女人名字,含着柔弱。

1.2.2 Ruby 的优点

用 Ruby 可以做桌面应用开发,也可以做 Web 开发。与 Ruby 类似的语言还有 Perl,Python,PHP 和 Java。Ruby 在这些语言中可以排在第三,虽然当不了第一,也算后来居上了。写语言不难,写好却不容易。Ruby 现在还有问题,有许多不完善的地方,许多地方都在改进,所以,它是一门处于发展中的语言。Ruby 的特点如下。

(1) 解释执行

Ruby 是解释型语言,因此不需编译即可运行,这样就提高了调试的速度。当然,解释型语言也存在执行效率低的问题。

(2) 变量无类型

Ruby 的变量没有类型,相应地,错误检查功能也变弱了。Ruby 的标识名区分大小写。

(3) 变量不需要声明

所有变量均无需声明即可使用。

(4) 语法简单

Ruby 语法比较简单,类似 Algol 系语法。

(5) 不需要内存管理

具有垃圾回收(garbage collect,GC)功能,能自动回收不再使用的对象。

(6) 纯粹的面向对象

Ruby 是纯粹的面向对象语言,因此包括整数等基本数据类型都是对象,它们都有发送信息的统一接口。在 Java 里面,基本类型和类是不一样的(如:int 是基本类型,Integer 则是类)。

(7) 迭代器

迭代器功能可以将循环抽象化,从而使得代码简洁。

(8) 功能强大的字符串操作

Ruby 以 Perl 为样板创造出了功能强大的字符串操作和正则表达式检索功能。

(9) 超长整数

添加超长整数功能后,可以计算非常大的整数。例如计算 400 的阶乘也轻而易举。

(10) 可以直接访问操作系统

Ruby 可以使用(UNIX 的)绝大部分的系统调用。单独使用 Ruby 也可以进行系统编程。

(11) 特殊方法

可向某对象添加方法。

(12) 用模块进行混合插入

Ruby 故意舍弃了多重继承,但拥有混合插入功能。使用模块来超越类的界限以共享数据和方法等。

(13) 闭 包

可以将某过程片段对象化。对象化后的该过程片段就称作闭包。

(14) 功能强大的字符串操作/正则表达式

Ruby 以 Perl 为样板创造出了功能强大的字符串操作和正则表达式检索功能。

(15) 具有错误处理功能

错误处理功能使得人们可以编写代码处理出错情况。

(16) 动态加载

若操作系统支持,可以在运行时读入对象文件。

Ruby 将许多常用的功能以方法的形式提供了出来,这使得在很多时候都不需要去编写函数来实现需要的功能。许多人使用过 Ruby 之后,都发现编程的效率大大提高了。使用 Ruby 编程很快,开发效率高,程序员就可以提高工作效率。于是这语言越来越流行。

1.3 Ruby 的安装

1.3.1 下载 Ruby

从 RubyForge 网站中,可以下载到与 Ruby 开发相关的软件。进入 <http://rubyforge.org/> 网站,如图 1-2 所示。

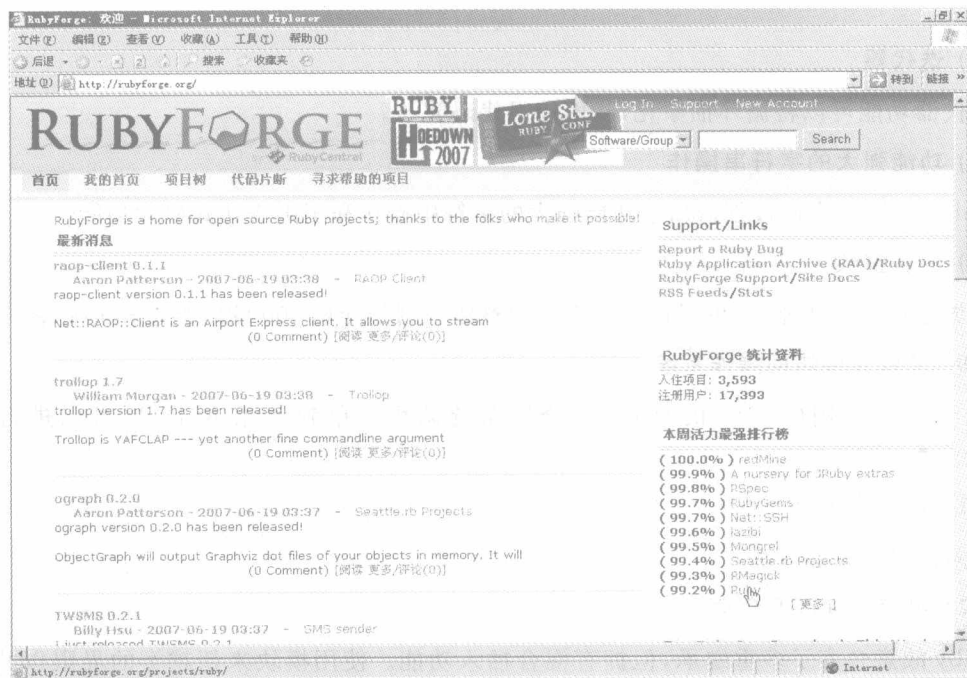


图 1-2 RubyForge 网站

进入 <http://rubyforge.org/projects/ruby/>, 可以下载 Ruby, 如图 1-3 所示。

单击“下载”按钮, 页面转入 Ruby 不同版本的下载列表, 如图 1-4 所示。

为了安装 Ruby, 需要下载的是 Ruby Installer, 这需要进入 <http://rubyforge.org/projects/rubyinstaller/>, 如图 1-5 所示。

单击“下载”按钮, 页面转入 Ruby Installer 不同版本的下载列表页面, 如图 1-6 所示。

当前(2007-6-22)最新版本的 Ruby Installer 是 186-25, 将其下载, 结果如图 1-7 所示。