

## <<Ruby入门权威经典>>

### 图书基本信息

书名：<<Ruby入门权威经典>>

13位ISBN编号：9787811245769

10位ISBN编号：7811245760

出版时间：2009-4

出版时间：北京航空航天大学出版社

作者：于天恩

页数：251

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>

## <<Ruby入门权威经典>>

### 前言

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 ( ) n 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？

国人翻译韵不好吗？

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。

## <<Ruby入门权威经典>>

### 内容概要

本书基于Ruby的18625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。

## &lt;&lt;Ruby入门权威经典&gt;&gt;

## 书籍目录

|                  |                |                |                |                 |              |                |                |               |              |                      |                       |           |              |                       |          |              |                |              |             |             |           |            |               |            |             |                |             |               |               |             |          |            |            |            |       |            |              |             |         |             |             |          |              |                 |    |       |
|------------------|----------------|----------------|----------------|-----------------|--------------|----------------|----------------|---------------|--------------|----------------------|-----------------------|-----------|--------------|-----------------------|----------|--------------|----------------|--------------|-------------|-------------|-----------|------------|---------------|------------|-------------|----------------|-------------|---------------|---------------|-------------|----------|------------|------------|------------|-------|------------|--------------|-------------|---------|-------------|-------------|----------|--------------|-----------------|----|-------|
| 第一部分 Ruby简介和环境配置 | 第1章 配置Ruby运行环境 | 1.1 面向对象和MVC模式 | 1.1.1 面向对象     | 1.1.2 MVC模式     | 1.2 Ruby概述   | 1.2.1 Ruby的由来  | 1.2.2 Ruby的优点  | 1.3 Ruby的安装   | 1.3.1 下载Ruby | 1.3.2 安装Ruby         | 小结                    | 思考和练习     |              |                       |          |              |                |              |             |             |           |            |               |            |             |                |             |               |               |             |          |            |            |            |       |            |              |             |         |             |             |          |              |                 |    |       |
| 知识               | 第2章 Ruby语言基础   | 2.1 最基本的知识     | 2.1.1 基本输出     | 2.1.2 引号的用法     | 2.1.3 转义字符输出 | 2.1.4 连句和换行    | 2.1.5 连行符      | 2.1.6 注释符     | 2.1.7 局部变量   | 2.1.8 基本输入           | 2.1.9 数字和字符串连接        | 2.2 类     | 2.2.1 最简单的类  | 2.2.2 类的基本使用          | 2.2.3 继承 | 2.2.4 单态方法   | 2.2.5 方法访问控制   | 2.2.6 属性读写控制 | 2.2.7 垃圾收集  | 2.2.8 异常处理  | 2.3 常量和变量 | 2.3.1 常量   | 2.3.2 全局变量    | 2.3.3 实例变量 | 2.3.4 局部变量  | 2.4 运算         | 2.4.1 算术运算符 | 2.4.2 关系运算符   | 2.4.3 逻辑运算符   | 2.4.4 其他运算符 | 2.5 流程控制 | 2.5.1 顺序结构 | 2.5.2 选择结构 | 2.5.3 循环结构 | 2.6 块 | 2.6.1 块的概念 | 2.6.2 块的基本使用 | 2.6.3 带参数的块 | 2.7 迭代器 | 2.7.1 迭代的概念 | 2.7.2 编写迭代器 | 2.8 过程对象 | 2.8.1 创建过程对象 | 2.8.2 把过程对象当作参数 | 小结 | 思考和练习 |
| 第3章 Ruby的数据类型    | 3.1 数字         | 3.1.1 数字的基本使用  | 3.1.2 数字的常用方法  | 3.1.3 数学计算方法    | 3.2 字符串      | 3.2.1 字符串的基本用法 | 3.2.2 字符串的常用方法 | 3.2.3 字符串方法总结 | 3.3 正则表达式    | 3.3.1 Ruby正则表达式的基本用法 | 3.3.2 正则表达式在字符串函数中的使用 | 3.4 日期和时间 | 3.4.1 Time对象 | 3.4.2 Date和DateTime对象 | 3.5 散列表  | 3.5.1 散列表的构造 | 3.5.2 散列表的常用方法 | 3.6 区间       | 3.6.1 区间的概念 | 3.6.2 区间的使用 | 3.7 数组    | 3.7.1 构造数组 | 3.7.2 数组的主要方法 | 3.8 结构体    | 3.8.1 建立结构体 | 3.8.2 结构体的主要方法 | 3.9 数据类型转换  | 3.9.1 通用的转换方法 | 3.9.2 自定义转换方法 | 小结          | 思考和练习    |            |            |            |       |            |              |             |         |             |             |          |              |                 |    |       |
| 第4章 模块和线程        | 4.1 模块         | 4.1.1 模块的概念    | 4.1.2 Mixin    | 4.1.3 Ruby的命名约定 | 4.2 线程       | 4.2.1 线程的概念    | 4.2.2 线程的同步    | 小结            | 思考和练习        |                      |                       |           |              |                       |          |              |                |              |             |             |           |            |               |            |             |                |             |               |               |             |          |            |            |            |       |            |              |             |         |             |             |          |              |                 |    |       |
| 第三部分 Ruby的高级知识   | 第5章 文件和目录      | 第6章 Ruby的数据库操作 | 第7章 桌面应用和Web开发 |                 |              |                |                |               |              |                      |                       |           |              |                       |          |              |                |              |             |             |           |            |               |            |             |                |             |               |               |             |          |            |            |            |       |            |              |             |         |             |             |          |              |                 |    |       |

## &lt;&lt;Ruby入门权威经典&gt;&gt;

## 章节摘录

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 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RedStone(红石头)。

这个名字长,而且生硬,不太好听,所以改为R11by(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。

## <<Ruby入门权威经典>>

(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。

<<Ruby入门权威经典>>

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>