

Elixir 程序设计



[Elixir 程序设计 下载链接1](#)

著者:Dave Thomas

出版者:电子工业出版社

出版时间:2016-3

装帧:平装

isbn:9787121282645

在了解 Elixir 前，在网上看到一些评论说，Elixir 其实就是用 Ruby 的语法来编写 Erlang 程序。其实不然。真正玩了 Elixir 之后，就会发现这纯属幻觉。因为 Elixir 相对于 Ruby

使用了截然不同的语言范式——函数式编程。两者在编程思想上相距甚远。如果说它看似 Ruby，也仅仅是 Elixir 设计者 José Valim，一个 Ruby 开发者的情怀使然罢了。Elixir 只是在 Erlang 之上加入的一层语法糖，在不改变 Erlang 开发实践的基础上，还能重用已有的 Erlang 代码。然而这层小小的话语糖却给内心坚若磐石但语法苦涩难懂的 Erlang 增添了一副平易近人的外表。虽然函数式编程并非新鲜事物，而 Elixir 则让函数式编程更现代化。

原著作者就是 15 年前编写了 Programming Ruby，将 Ruby 带入大众视野的 Dave Thomas。他延续了一贯的写作风格，以一个拥有面向对象的开发经历但并未接触过函数式编程的开发者的角度切入，循序渐进地带领读者进入 Elixir 的奇妙世界。通过一步步的实验探索，引导读者逐步跳出已有的编程思维模式，以全新的函数式编程方式来思考及寻找解决实际问题的办法。然而本书并不像 Programming Ruby 那样面面俱到，而是通过给读者讲述基本的 Elixir 语法和编程思想，同时提供许多有用的资源，启发读者继续深入探索。这才是学习一门新语言的乐趣所在。

作者介绍:

原著作者就是 15 年前编写了 Programming Ruby，将 Ruby 带入大众视野的 Dave Thomas。他延续了一贯的写作风格，以一个拥有面向对象的开发经历但并未接触过函数式编程的开发者的角度切入，循序渐进地带领读者进入 Elixir 的奇妙世界。通过一步步的实验探索，引导读者逐步跳出已有的编程思维模式，以全新的函数式编程方式来思考及寻找解决实际问题的办法。然而本书并不像 Programming Ruby 那样面面俱到，而是通过给读者讲述基本的 Elixir 语法和编程思想，同时提供许多有用的资源，启发读者继续深入探索。这才是学习一门新语言的乐趣所在。

目录: 第 1 章 接受现实
编程时应该关注数据转换
借助管道来组合转换
函数是数据转换器
安装 Elixir
运行 Elixir
iex—交互式 Elixir
编译和运行
对阅读本书的建议
练习
换一种方式思考
第一部分 常规编程
第 2 章 模式匹配
赋值：并非如你所料
更复杂的匹配
轮到你了
用 _ (下画线) 忽略匹配值
每次匹配变量仅绑定一次
轮到你了
从另一个角度来看等号第 3 章 不可变性
你已经拥有了 (一些) 不可变数据
不可变的数据才是已知的

不可变性对性能的影响

复制数据

垃圾回收用不可变数据编写程序

第 4 章 Elixir 基础

内置类型

值类型

整数

浮点数

原子

区间

正则表达式

系统类型

PID 和 端口

引用

收集类型

元组

列表

散列表

二进制型

命名、源文件、约定、运算符和其他

真值

运算符

小结

第 5 章 匿名函数

函数和模式匹配

轮到你了

一个函数，多个函数体

编写更长的代码

轮到你了

能返回函数的函数

记住原始环境的函数 参数化函数

轮到你了

将函数作为参数来传递

& 运算符

轮到你了

函数是核心

第 6 章 模块与命名函数

编译模块

函数体是代码块

轮到你了

函数调用与模式匹配

轮到你了

哨兵子句

哨兵子句的限制

默认参数

轮到你了

私有函数

|> — 美妙的管道运算符

模块

模块指令

import 指令

alias 指令

require 指令

模块属性

模块名：Elixir、Erlang和原子类型

调用 Erlang 的库函数

寻找函数库

轮到你了

第 7 章 列表与递归

头部和尾部

使用头部和尾部来处理列表

iex 如何显示列表

使用头部和尾部来构造列表

创建映射函数 在递归过程中跟踪值

轮到你了 生成求和函数

轮到你了

更复杂的列表模式

列表的列表

轮到你了

List 模块实战

与列表友好相处

第 8 章 字典：散列表、散列字典、关键字列表、集合与结构体

如何在散列表、散列字典和关键字列表之间做选择

字典

模式匹配和更新散列表

模式匹配不能绑定键

更新散列表

散列表与结构体

访问结构体的另一种方式

嵌套字典结构

嵌套访问器和非结构体

动态（运行时）嵌套访问器

集合

能力越大，诱惑越大

第 9 章 番外篇—类型是什么

第 10 章 处理收集—Enum 与 Stream

Enum——处理收集

关于排序的说明

轮到你了

Stream——延迟处理的枚举类型

流是可组合的枚举器

无限流

自定义流

流在实际中的应用

Collectable 协议

推导式

推导式也可以处理二进制位

作用域与推导式

推导式的返回值

轮到你了 感动过往的神

第 11 章 字符串与二进制型

字符串字面量 Heredoc 魔术符

“字符串”这个名称

单引号字符串 —— 字符编码列表 轮到你了

二进制型

双引号字符串是二进制型

字符串与 Elixir 库

轮到你了

二进制型与模式匹配

用二进制型来处理字符串

轮到你了

熟悉却又陌生

第 12 章控制流

if 与 unless

Cond

Case

抛出异常

包含异常的设计

四两拨千斤

轮到你了

第 13 章组织项目

项目：从 GitHub 获取 issue

我们的代码将如何工作

任务：用 mix 来创建我们的新项目

创建项目树

转换：解析命令行

进阶：编写一些基本的测试

轮到你了

转换：从 GitHub 获取数据

任务：使用外部库

寻找库

为项目添加库

轮到你了

回到转换

转换：转换响应内容

不在 hex 里的依赖项

应用程序配置

转换：为数据排序

转换：取前 n 条

轮到你了

转换：格式化表格

任务：创建命令行可执行程序

任务：添加日志

任务：测试代码注释

任务：创建项目文档

使用转换数据的方法来编写代码

轮到你了

第二部分 并发编程

第 14 章运用多进程

简单的进程

在进程间发送消息

处理多条消息

递归、循环与栈

进程开销

轮到你了

进程何时结束

关联两个进程

监控进程

轮到你了

并行 map——Erlang 版本的 “Hello, World”

轮到你了

斐波那契数服务器

任务调度器
轮到你了
代理——一个难题
以进程的方式来思考
第 15 章 节点——分布式服务的关键
命名节点
轮到你了
节点、cookie 与安全
给你的进程命名
何时给进程命名
轮到你了
I/O、PID 与节点
轮到你了
节点是分布式的基础
第 16 章 OTP：服务器
一些 OTP 的定义
OTP 服务器
状态和单个服务器
我们的第一个 OTP 服务器
轮到你了
单向调用
跟踪服务器的执行情况
轮到你了 GenServer 回调函数
给进程命名
整理接口
轮到你了
第 17 章 OTP：应用程序监视器
应用程序监视器与工作进程
轮到你了
管理重启前后的进程状态
应用程序监视器是可靠性的核心
轮到你了
第 18 章 OTP：应用程序
这不是传统的应用程序
应用程序规范文件
将 Sequence 程序转变为 OTP 应用程序
关于应用参数的更多信息
监视是可靠性的基础
轮到你了 热代码交换
OTP 很大——难以置信的大
轮到你了
第 19 章 任务与代理
任务
任务与监视
代理
更大的例子
使其分布式运行
使用代理与任务，还是 GenServer
第三部分 更高级的 Elixir
第 20 章 宏与代码求值
实现 if 语句
宏注入代码
装载次序 quote 函数
将内部表示作为代码使用

unquote函数
展开列表——unquote_splicing
回到我们的 myif宏
轮到你了
使用绑定来注入值
宏是卫生的
执行代码片段的其他方法
宏与运算符
深入研究
更进一步 轮到你了
第 21 章 连接多个模块：行为与 use
行为
定义行为
声明行为
use 与 using
放到一起——跟踪方法调用
使用 use 轮到你了
第22章协议 ——多态函数
定义协议
实现协议
可用的类型
轮到你了
协议和结构体
内置协议： Access
内置协议： Enumerable
内置协议： StringChars
内置协议： Inspect
协议就是多态
轮到你了
第 23 章 更酷的玩意儿
自定义魔术符
获取选项
轮到你了
多应用的 umbrella 项目
创建 umbrella 项目
创建子项目 LineSigil 项目
Evaluator 项目
连接子项目
别急！还有更多炫酷的玩意儿！
附录 A 异常： raise 与 try, catch 与 throw
附录 B 类型规范与类型检查
· · · · · (收起)

[Elixir 程序设计 下载链接1](#)

标签

Elixir

Erlang

程序设计

编程

Functional-Programming

programming

elixir

Ruby

评论

是也乎 (╯▽╰) 张海龙都看不懂的书 --> 哈哈哈哈哈哈哈不死药就是屌 --> 上心了 实话 ruby 畅销书作者来写并不是什么优势啊 --> 没有趣学 haskell和erlang 那两本小书好看 --> 也就后面下载 github issue 的实例接入现实世界了 --> 讲真 OTP 合用场景还没有冲出电信和游戏~并不是本身能力 --> 而是组成 生态 的人啊 <-- 都在窄小领域中~ 好在逆天的前沿设计一点不落后时代 --> 特别是多核量子架构 --> actor 天然兼容…就等大厂带头了 --> 咱们先用起来~

@ memect

跟官方文档比起来有点写的不够细，语言属于也没有那么生硬！

后面几章翻译不是很流畅...
另外本人对Erlang和其OTP平台不熟，后面几章看起来比较吃力...
宏什么的看起来好混乱的感觉... 本书貌似不太适合 Elixir 入门。

为什么字号要变大，用比较正常的字号不行吗？无语

第13章的实例正是与真实世界的连接，这才是编程的正确姿势，满分

其实我觉得读官网英文教程就够了...

就翻译来看还是不错的，看得出译者是多年的开发者，表达得通俗易懂。内容不用说了，Elixir是集语言之大成者。对比Java 8中引入的语言新语言特性主要有三个：函数参数化，Stream API，接口默认实现。函数参数化就是函数式语言的基本特性，Stream API是可以把数据处理理解成Unix pipe。这些特性在Elixir是原生的设计，优雅得多。除此之外Elixir还引入了Java想引入而未成的函数调用模式匹配。并发方面自不用说，基于Erlang的高并发模式无论在语法表达还是在运行效率方面是非常经得住考验的。总之，无论是技术老鸟还是技术新人，这本书都很有价值，开阔视野，触发重现思考编程的方式。

语言非常流畅的一本编程书，看完后感觉很爽。

这本确实能开拓视野，函数式编程应该是未来的趋势。

这本书是CODING团队的人翻译的，这是个有追求的技术团队。用过Coding WebIDE的人都知道这也是CODING团队的良心作品。而IDE里面就用Elixir实现WebIDE的sharding子服务，让sharding子服务可以不停机升级。总而言之，推荐！

这本书翻译的不错，写的比较浅显，Dave写书的功力还是不错的，如耳边低语，逻辑清楚，会把关于语言的爽点告与你知。elixir语言是ruby+erlang+lisp的完美混合体，既有ruby的一致爽利，还有erlang的并发高活，更有lisp的灵活和高扩展。语言的威力无与伦比，同时又保持了fp的简洁理念，基本不用记忆语法，熟悉表达后靠猜就能写代码了。深厚的底蕴绝对值得一学一用。5星推荐！

[Elixir 程序设计 下载链接1](#)

书评

或许，你看了下面的书评会觉得我是在帮忙打广告的，但是我只想说，我仅以一个2年coding.net 用户来评价一下这本书 :)

正如这本书的推荐序一样，看了elixir，的确让人感到elixir脑洞大开，一年前搞过一段时间erlang，然而由于社区的不够活跃，又碰到了chicago这个w...

Elixir 之前一直只有英文版本。后来看到杜叔着手翻译了这边 Elixir 算是中文版本的开荒之作，好像在 WebIDE 里面也用到了。 Elixir 是一门暴力语言，基于 Erlang，天然处理并发的。对于书还没来得及全部读完，但是对于译者我是很是熟知的。 Coding WebIDE 的架构师杜万，杜叔...

[Elixir 程序设计 下载链接1](#)