

花二十分钟教你Ruby快速入门（四）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E8_8A_B1_E4_BA_8C_E5_8D_81_E5_c104_145565.htm 保存上面的代码到名为“ri20min.rb”的文件中，并使用“ruby ri20min.rb”的命令执行它。程序输出如下：下面我们将深入了解一下上面的代码。请注意上面代码中的起始行，它以#开头。在Ruby语言中，任何以#开头的行都被视为注释，并被解释程序忽略。我们的say_hi方法已经发生了变化：它查找@names参数并按照其参数值作出决定：如果参数值为nil，它将打印三个圆点。那么@names.respond_to?("each")表示什么？循环也叫迭代。如果@names对象具有each方法，那么它是可以被迭代的，进而可以对其进行迭代，从而问候列表中每个人。如果@names不具备each方法，则将它自动转换为字符串，并执行默认的问候。each是一种方法，它接受一个代码块（block of code），然后针对列表中的每个成员执行这个代码块，而在do和end之间的部分便是这个非常类似匿名函数的代码块。在管道符之间的变量是代码块的参数name，它作为代码块参数被绑定为列表成员，而代码块puts "Hello #{name}!"将使用这个参数进行输出。大多数其它的编程语言使用循环遍历列表，下面是C语言的循环示例：上面的代码显然可以工作，但它不够“优雅”！你不得不用i这个多余的循环变量，还需要指出列表的长度，然后再解释如何遍历列表。Ruby的迭代方式则更加优雅，所有的内部管理细节都隐藏在each方法中，你所需做的就是告诉它如何处理其中的每个成员。块(block)，Ruby边缘的高亮点！块(block)的真正优势在于：能够处理比列表

更加复杂的对象。除了在方法中可以处理简单的内部管理细节外，你还能处理setup、teardown和所有错误，而不让用户有所察觉。say_bye方法没有使用each，而是检查@names是否具有join方法，如果具有join方法，则调用join方法。否则它将直接打印@names变量。此方法并不关心变量的实际类型，这依赖于它所支持的那些被称为“Duck Typing”的方法：duck typing是动态类型的一种形式：变量的值自身隐含地决定了了变量的行为。这暗示了某个对象与其它实现了相同接口的对象之间是可交换的，不管对象之间是否具有继承关系。鸭子测试（duck test）是对duck typing的一种形象比喻“如果它走路像鸭子，那么也一定像鸭子一样呷呷地叫，那么它必定是一只鸭子”。duck typing是某些编程语言的特性：如Smalltalk, Python, Ruby, ColdFusion。Duck Typing的益处是无需对变量的类型进行严格地限制，如果某人使用一种新类型的列表类，只要它实现了与其它列表相同语义的join方法，便可以拿来使用。启动脚本文件上半部分是MegaGreeter类的代码，而后面剩下的部分则是对这些类方法的调用。而这是我们最后值得注意的一点：__FILE__是一个“具有魔力”的变量，它代表了当前文件名。\$0是用于启动程序的文件名。那么代码“if __FILE__ == \$0”便意味着检查此文件是否为将被使用的主程序文件。这样做可以使程序文件作为代码库使用，而不是可执行代码；但当此文件被用作执行文件时，也可被执行。如何进一步学习Ruby到此便是本入门的尾声了。当然还有许多值得浏览的：Ruby提供的各种不同的控制结构；块和yield的使用；模块作为mixins使用等。希望这次Ruby初体验能使你对Ruby更感兴趣。注：mixin在面向对象编程语言中是一种提

供某些功能给子类继承的类，但mixin并不能实例化。从某个mixin继承并不是什么特殊的形式，而它更适于收集功能。某个子类甚至可以通过继承一个或者多个mixin选择继承它的全部或者多数功能。一个mixin能延期定义和绑定方法直到运行时，而属性和实例参数也将在编译时才被定义。这不同于多数常见的方式：定义所有的属性、方法，并在编译时进行初始化。如果这样的话，请埋头翻阅我们的文档，那里有免费、丰富的在线手册和入门资源。或者如果你喜欢在啃书本的话，可以到图书列表中选择一些你所需要的。100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com