Ruby on Rails DEFINITION: Ruby on Rails is a software development environment that is built with and uses the object-oriented programming language Ruby, with the overall aim of making programmers ’ jobs both more fun and more productive. The programming language Ruby has been around since 1993. Initially popular in Japan, its use has been growing and widening. Ruby got a big boost in 2004 with the release of a new programming environment called Rails that was built around Ruby. Before discussing Rails, let ’ s first examine Ruby and see what makes it different from other languages. Ruby is a pure object-oriented, open-source programming language with a very clean syntax that one writer has said “ combines Smalltalk ’ s elegance, Python ’ s ease of use and Perl ’ s pragmatism. ” An interpreted scripting language, Ruby is as useful for creating small ad hoc scripts as it is for full-scale applications. Using a direct-execution model and dynamic typing, Ruby lets you develop code incrementally. in most cases, you can add a feature and then try it immediately. Ruby programs are usually shorter than their Perl, Python or C counterparts. Ruby ’ s creator, Yukihiro Matsumoto, has said he wanted a language that would make him more productive and at the same time be fun to use. In fact, this desire for fun in programming is probably Ruby ’ s biggest single point of differentiation from any other language. When one first starts reading the Ruby literature, it ’ s easy to think that it ’ s a boutique language with a cultlike following of near-fanatic users. But remember that Perl, PGP and Python started out in much the same way. Every language is designed with specific purposes: ease of coding. compactness of code. readability. speed of execution. orientation for specialized situations such as text processing, database handling or numerical computation. Even so, with most languages that are Turing-complete, you can pretty much do anything you need. the practical differences from one to another will be in the areas of form, style, size, ease of use or maintainability. For example, Python and Ruby provide almost the same power to the programmer. Ruby ’ s syntax is such that once you ’ re familiar with its vocabulary and style, you can read many lines of code in something like natural language, making programs much easier to understand and maintain. Ruby eliminates the need to start out each program with long statements declaring structures and variables. Like Perl, Ruby has features that make it easily adaptable to processing text files and to systems management tasks. You can add methods to a class, or even to an instance during runtime, so that you can have an instance of one class behave differently from other instances of the same class. Also like Perl, but quite different from Python, Ruby lets you code things in many ways, thus it can appeal to programmers with very different styles and personalities. Rails is a full-stack, open-source programming framework implemented in Ruby and designed for writing database-oriented real-world Web applications quickly and easily. It was created by David Heinemeier Hansson, a

Danish college student, based on a project management tool he ’ d written called Basecamp. Ruby on Rails was released to the public in July 2004. By full-stack framework, Hansson means that all layers are built to work together so that you don ’ t repeat yourself (DRY) and you can use a single language from top to bottom. Everything in Rails is written in Ruby. To achieve DRY, Rails relies on a couple of guiding principles. The first is less software, meaning you write fewer lines of code. Keeping code small speeds development and makes for fewer bugs, and it makes code easier to understand, maintain and enhance. One feature of Ruby that contributes to writing less code is that it assumes that the framework itself can and should handle metadata. The second is the principle of convention over configuration, meaning that Rails doesn ’ t use configuration files, relying instead on some simple programming conventions that let it figure out what it needs to know by using information already contained in the application code and the database that ’ s running. Ruby on Rails 定义：Ruby on Rails是一种软件开发环境，它是利用面向对象编程语言Ruby构建起来的，其总的目标就是要使程序员的工作比较有趣、生产率也较高。 编程语言Ruby自1993年起就已存在。最初，它在日本很流行，其应用在扩大。在2004年，随着一个叫Rails、围绕Ruby构建起来的新颖编程环境的问世，Ruby获得了一次大发展。在讨论Rails之前，先让我们看看Ruby,它是如何使其有别于其他的语言。 Ruby是一种纯粹的面向对象的开放源码的编程语言，它有非常清晰的语法，有位作者称，它将Smalltalk的精致、Python的易用和Perl的实用结合一身。作为一种解释型的脚本语言，Ruby对

生成短小的特定脚本与编写完整的应用程序一样有用。利用直接执行模型和动态键入，Ruby让你用逐步增加的方式开发程序，在多数情况下，你能增加一个功能，并能马上进行试验。Ruby程序通常比同类的Perl、Python、或C 程序都要短。Ruby的发明人Yukihiro Matsumoto说过，他需要一种既能使他有较高的生产率，同时又能乐意使用的语言。事实上，让编程好玩的想法可能是Ruby与其他任何一种语言最大的不同之处。当一个人首次阅读Ruby文献时，很容易就认为它是一种热门的语言，有着一群近乎狂热的用户崇拜它追随它。但请记住，Perl、PGP和Python开始时也有类似的情况。 每一种语言都是为特定目的设计的：易于编程、程序紧凑、可读性、执行速度、面向特定情形（如文本处理）、数据库处理或数值计算。即便这样，对于大多数具有完整图灵原理的语言来说，你能用它们做你需要做的任何事，它们之间实际上的差异在于形式、风格、大小、易用性和可管理性。例如，Python和Ruby给程序员提供了几乎是相同的功能。 Ruby的语法就是这样的：一旦你熟悉它的词汇和风格，你就能像读自然语言那样一目几行，使得程序更容易理解和维护。Ruby不需要在开始每个程序时先写很长的注释语句，说明结构和变量。 Ruby像Perl那样具有不少功能，使之很容易适应处理文本文件和系统管理等任务。你能给类增加方法，甚至在运行中给实例增加方法，从而你就能拥有不同于同一类的其他实例的类行为实例。Ruby还能像Perl那样、但不同于Python，让你用多种方法对事情编程，因而它能以不同的风格和个性吸引程序员。 Rails是用Ruby实现的全堆栈、开放源码的编程架构，旨在快速、简易地编写面向数据库的、真实世界

的Web应用程序。它是由一名丹麦的大学生David Heinemeier Hansson发明的，它基于一个由他编写的称作Basecampde的项目管理工具。Ruby on Rails是在2004年7月公开的。所谓全堆栈架构，Hansson是要构建所有层面，以便一起工作，从而你不必重复你自己的工作（DRY），从顶到底都能用同一语言。Rails中所有东西都是用Ruby写的。为了获得DRY，Rail有两条指导原则。第一条是较小的软件，这意味着你编写较少的程序行。程序小就能加速开发，错误也较少，使得程序较容易理解、维护和增强。对编写较小程序做出贡献的Ruby的一个特性就是它假设该架构本身能够并应该处理元数据。第二条是对配置的协定原理，这意味着Rails不用配置文件，而是依靠一些简单的编程协定，让它通过利用应用程序已有的信息和在运行的数据库，解决它需要知道什么这个问题。