Introduction

I wanted to minimize my frustration during programming, so I want to minimize my effort in programming. That was my primary goal in designing Ruby. I want to have fun in programming myself.

—Yukihiro Matsumoto (Matz), creator of Ruby

Ruby is a "best of breed" language that has been assembled from the best and most powerful programming features found in its predecessors.

—Jim White

Ruby makes me smile.

—Amy Hoy (slash7.com)

Ruby is a fun toy. It's also a serious programming language. Ruby is the jolly uncle who keeps the kids entertained, but who puts in solid 12-hour days at the construction site during the week. To hundreds of thousands of programmers, Ruby has become a good friend, a trusted servant, and has revealed a new way of thinking about programming and software development.

Like the guitar, it's often claimed that Ruby is an easy language to learn and a hard one to master. I'd agree, with some provisions. If you don't know any programming languages already, Ruby will be surprisingly easy to learn. If you already know some languages such as PHP, Perl, BASIC, C, or Pascal, some of the concepts in Ruby will already be familiar to you, but the different perspective Ruby takes with problem solving will probably throw you at first. Like the differences between spoken languages, Ruby differs from most other programming languages not only by syntax, but by culture, grammar, and customs. In fact, Ruby has more in common with more esoteric languages such as LISP and Smalltalk than with better-known languages such as PHP and C++.

While Ruby's roots might be different from other languages, it's heavily used and respected in many industries. Companies that use or support Ruby in one way or another include such prestigious names as Sun Microsystems, Intel, Microsoft, Apple, and Amazon.com. The Ruby on Rails Web framework is a system for developing Web applications that uses Ruby as its base language, and it powers hundreds of large Web sites. Ruby is also used as a generic language from the command prompt, much like Perl.

Grammarians, biochemists, database administrators, and thousands of other professionals and hobbyists use Ruby to make their work easier. Ruby is a truly international language with almost unlimited application.

This book is designed to cater both to people new to programming and those with programming experience in other languages. Ruby's culture is different enough from other languages that most of this book will be of use to both groups. Any large sections that can be skipped by already proficient programmers are noted in the text. In any case, I'd suggest that all programmers at least speed-read the sections that might seem obvious to them, as there are some surprising ways in which Ruby is different from what you've done before.

When reading this book be prepared for a little informality, some quirky examples, and a heavy dose of pragmatism. Ruby is an extremely pragmatic language, less concerned with formalities and more concerned with ease of development and valid results. From time to time I'll show you how you can do things the "wrong" way in Ruby, merely for illustrative purposes, but mostly you'll be working with code that does things "the Ruby way." When I started to learn Ruby I learned primarily by example, and with a language as original and idiomatic as Ruby, it's the easiest way to pick up good habits for the future. However, there's always "more than one way to do it," so if you think some code in this book could be rewritten in a different way that fits in more with your way of thinking, try it out!

As you start this book, be prepared to think in new ways, and to feel motivated to start coding for both fun and profit. Ruby has helped a lot of jaded developers become productive once again, and whether you're a beginner to programming or one of those jaded programmers, it's almost inevitable that you'll see how Ruby can be both fun and productive for you.

Last, if you're coming from other modern scripting languages such as Perl, PHP, or Python, you might want to jump to Appendix A before reading Chapter 1. It covers the key differences between Ruby and other scripting languages, which might help you move through the initial chapters of this book more easily.

Good luck, and I hope you enjoy this book. I'll see you in Chapter 1.