To: Jessica Biron <j.biron@neu.edu>

From: Isaac Boehman <isaac@boehman.me>

Subject: Favorite Programming Language

Date: January 23, 2015

The Ruby programming language has a distinct design philosophy. During a 2008 Google Tech Talk [1], Yukihiro "Matz" Matsumoto (creator of Ruby), stated "I hope to see Ruby help every programmer in the world to be productive, and to enjoy programming, and to be happy. That is the primary purpose of Ruby language."

Ruby ships with a REPL, which eases learning the language by being able to directly interact with it, allowing for rapid prototyping and quickly testing out new ideas. Functional-style programming is also supported. Everything is either an expression or an assignment and there's support for anonymous functions, closures, and first-class continuations. Additionally, every method implicitly recieves a block and can use it as it sees fit.

Along with this, Ruby has extensive support for metaprogramming. Everything inherits from the base `Object` and has a `self`. Every object is open, that is, any object can be arbitrarily extended. Class context are not special, meaning code can be evaluated inside a given class's context from anywhere. Additionally, the module system enables defining trait-like behaviors for objects, allowing behavior to be decoupled and reused. There is also support for various forms of syntactic sugar, such as parentheses not being required for method calls or definitions.

The features specified above allow for domain-specific languages to easily be implemented. Some examples of popular projects that are DSLs on top of Ruby include Chef and Puppet (configuration management), Sinatra and Ruby on Rails (web application frameworks), and Cucumber and RSpec (testing frameworks).

[1] https://youtube.com/watch?v=oEkJvvGEtB4