El lenguajge Ruby

Ariel Herrera
Saúl Zamora

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Lenguajes de Programación

September 24, 2016

Índice

Datos históricos

Ruby es un lenguaje de propósito general, dinámico, reflectivo y orientado a objetos. Fué diseñado y desarrollado a mediados de los años 90 por Yukihiro "Matz" Matsumoto en Japón.
De acuerdo con su creador, Ruby fue influenciado por Perl, Smalltalk, Eiffel, Ada y LISP. Soporta la programación en múltiples paradigmas incluyendo funcional, orientación a objetos e imperativo. También tiene un sistema de tipos dinámicos y un manejo de memoria automático.

Importancia y usos

Ruby es un lenguaje fácil de aprender, con fuertes abstracciones de los detalles computacioneles; dado esto es un buen lenguaje a considerar para comenzar a aprender a programar.
Ruby también es la entrada a Ruby On Rails. RoR es un framework con mucha popularidad que usa y depende de Ruby, dicho framework es ampliamente utilizado en la creación de aplicaciones web.

Tipos de datos

Ruby posee un tipado de datos dinámico, sin embargo posee los siguientes tipos de datos nativos:

- String
- Fixnum
- Integer
- Numeric
- Float
- NilClass
- Hash
- Symbol
- Array
- Range

Expresiones

- Operadores aritméticos:
  - +, -, *, /, %, ** (potencia)
- Operadores de comparación:
  - ==, !=, ¿, ¡, ¿=, ¡=, ¡=¿, ===, .eql?, equal?
- Operadores de asignación:
  - =, +=, -=, *=, /=, %=, **=
- Operadores lógicos:
  - and, or, &&, !, not

Estructuras de control

Estatuto IF:

```
# Generate a random number and
    print whether it's even or
    odd.
if rand(100) % 2 == 0
  puts "It's␣even"
else
  puts "It's␣odd"
end

puts "It's␣even" if rand(100) % 2
     == 0
```

Estatuto UNLESS:

```
# Generate a random number and
    print whether it's even or
    odd.
unless rand(100) % 2 == 0
  puts "It's␣even"
else
  puts "It's␣odd"
end
```

Bloques e iteradores:

```
{ puts 'Hello,␣World!' } # note
    the braces
# or:
do
  puts 'Hello,␣World!'
end

File.open('file.txt', 'w') do |
    file| # 'w' denotes "write
    mode"
  file.puts 'Wrote␣some␣text.'
end

    # file is automatically
    closed here

File.readlines('file.txt').each
    do |line|
  puts line
end
# => Wrote some text.
```

Características principales

- Fuerte orientación a objetos con herencia, mixins y metaclases.
- Tipado de datos dinámico y *duck typing* (si se ve como un pato y suena como un pato, es un pato).
- Todo es una expresión (hasta los estatutos) y todo se ejecuta imperativamente (hasta las declaraciones).
- Reflexión y alteración dinámica de objetos para facilitar metaprogramación.
- Sintáxis única de bloques para iteradores y generadores.
- Notación literal para arreglos, hashes, expresiones regulares y símbolos.
- Interpolación de hileras.
- Argumentos default.
- Recolector de basura.

Características distintivas

- Cuatro niveles de alcance para variables: globales, de clase, de instancia y local.
- Sobrecarga de operadores.
- Soporte nativo para números racionales, complejos y aritmética de presición arbitraria.
- Soporte inicial para Unicode y múltiples codificaciones de caractéres.
- Consola Ruby interactiva.
- Manejo centralizado de paquetes a travéz de *RugyGems*.
- Implementaciones en todas las plataformas más conocidas.

Ventajas

- Herramientas: utilizando Ruby On Rails, es posible entregar más *features* en menos tiempo. El framework provee de una estructura estándar para web apps.
- Disponibilidad de ibrerías.
- Comunidad grande y activa.
- Productividad: Ruby es un lenguaje elocuente, lo que combinado con la gran variedad de librerías externas, habilita al programador al desarrollo rápido.

Desventajas

- Velocidad de ejecución: el mayor argumento contra Ruby es que es lento. Sin embargo, estos problemas de velocidad no se notarán hasta que la aplicación tenga un gran volumen de tráfico.
- Velocidad de arranque: aplicable en el caso del framework, Ruby on Rails, dependiendo del número de dependencias (*gemas*), puede tomar una cantidad de tiempo significativa a una aplicación para arrancar, lo que resiente el tiempo de desarrollo.
- Documentación: puede ser difícil encontrar buena documentación, especialmente para las gemas y librerías que hacen uso excesivo de mixins.

Demostración

Gracias por su atención