

Ruby e a sua facilidade no aprendizado

Douglas P. Santos, Henrique O. de M. Silva

Centro Universitário de Anápolis - Unievangélica

Anápolis - GO - Brasil

douglas.santos@aluno.unievangelica.edu.br, henriqueolimpio.m@gmail.com

Abstract. *Ruby is a language with a proposal to be simple, easy to understand but also has many resources to help those who use it, being the great friend of the programmer, leaving some complicated parts for the computer to deal with. It tries to have an easy to read and write syntax.*

Keywords: *Ruby, Rails.*

Resumo. *Ruby é uma linguagem com uma proposta de ser simples, de fácil entendimento mas que também possui muitos recursos para ajudar quem a utiliza, sendo o grande amigo do programador, deixando algumas partes complicadas para o computador lidar. Procura ter uma sintaxe de fácil leitura e escrita.*

Palavras-Chave: *Ruby, Rails.*

1. Introdução

Ruby foi criado por Yukihiro “Matz” Matsumoto, se tornando público em 1995, ruby uniu partes das linguagens favoritas do seu criador (Perl, Smalltalk, Eiffel, Ada e Lisp), sendo mais poderosa do que Perl, e mais orientada a objetos do que Python. Foi aceito por programadores em todo mundo por ser uma nova linguagem que equilibra a programação imperativa com a programação funcional.

A fama da linguagem aumentou por volta de 2005 com o uso do Ruby on Rails um framework de aplicações web, que reúne as bibliotecas da linguagem. Nas palavras do seu criador Yukihiro Matsumoto “O Ruby é simples na aparência, mas muito complexo no interior, tal como o corpo humano”. A beleza do Ruby é encontrada em seu equilíbrio entre simplicidade e poder, sendo uma linguagem de fácil aprendizado com foco na simplicidade e produtividade, mas que dá uma grande liberdade aos desenvolvedores.

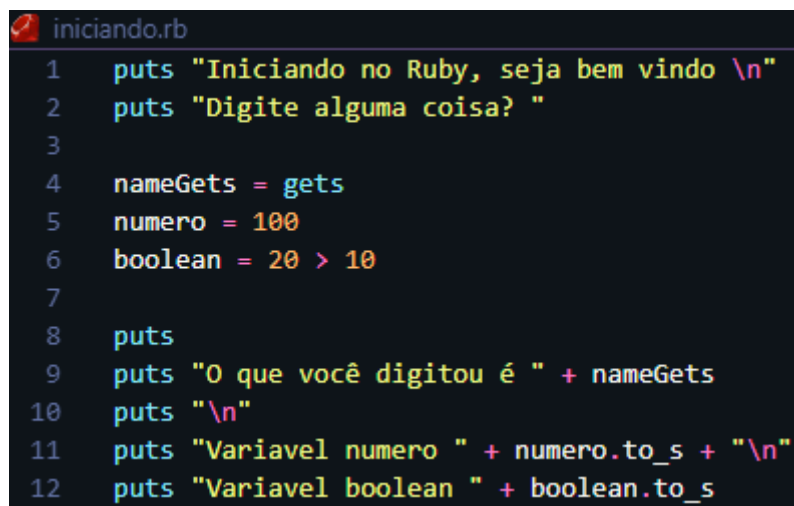
2. Desenvolvimento

Como em toda linguagem a primeira coisa que fazemos é escrever “Olá mundo”, em ruby a sintaxe é: `puts “Olá mundo”`. Para executar, utilize o terminal e digite na pasta onde esta arquivo: `ruby nomeDoArquivo.rb`. No seu console deverá aparecer Olá mundo.

2.1 Comandos simples

- Variável: Quando se deseja utilizar uma variável só precisa escrever o nome e atribuir um valor. Exemplo: `nome = “teste”`(variável tipo texto) ,`nome = 100`(variável tipo numérico), `nome = 100 > 150`(variável tipo boolean).No Ruby não precisa informar o tipo da variável, como no JavaScript.
- puts: É usado para escrever mensagem no console. Exemplo: `puts “Testando o puts”`, no console deve aparecer Testando o puts, só irá aparecer o texto dentro das “”, nesse caso. O puts só funciona com variáveis do tipo texto em uma concatenação. Exemplo:
`num = 12`
`puts "teste " + num.to_s`
- gets: É utilizado para capturar o que o usuário digita no console e atribuir a uma variável. Exemplo: `variavel = gets`.

Figura 1 - Primeiro programa



```
iniciando.rb
1 puts "Iniciando no Ruby, seja bem vindo \n"
2 puts "Digite alguma coisa? "
3
4 nameGets = gets
5 numero = 100
6 boolean = 20 > 10
7
8 puts
9 puts "O que você digitou é " + nameGets
10 puts "\n"
11 puts "Variavel numero " + numero.to_s + "\n"
12 puts "Variavel boolean " + boolean.to_s
```

Fonte: Autores.

Nesta imagem vemos os comandos dessa sessão e alguns comando a mais, em relação a imprimir algo no console, quando executamos esse trecho de código.

Linha 1: A novidade nessa linha é o `\n` ele é usado para quebra de linha, a próxima mensagem do console irá aparecer na linha abaixo. Caso queira pular várias linhas em sequência, pode utilizar o `\n` para fazer, quanto mais `\n` se coloca mais linhas serão puladas.

Linha 8: Quando se utiliza o `puts` sem colocar “” na frente, como na linha 8, no console ele simplesmente pula uma linha.

Linha 9: Caso precise mostrar um texto junto a uma variável usamos o `+` que é uma concatenação(juntar) o texto com a variável.

Linha 11: Se precisar mostrar um valor numérico ou booleano no puts, pode utilizar o `.to_s` para transformar a variável para o tipo texto, como é feito na linha 11 e 12. Executando o programa da imagem 1, o resultado será a imagem 2.

Figura 2 - Saída no console

```
PS C:\estudos\ruby\iniciando> ruby iniciando.rb
Iniciando no Ruby, seja bem vindo
Digite alguma coisa?
Estudano ruby

O que você digitou é Estudano ruby

Variavel numero 100
Variavel boolean true
```

Fonte: Autores.

2.2 Estrutura condicional

Quando precisamos verificar se uma condição ou operação é verdadeira ou falsa utilizamos o `if` em ruby e em diversas linguagens como C, Java, JavaScript, etc. O que muda de uma linguagem para outra é a sintaxe, mas todas se parecem bastante. Uma sintaxe simple do `if` em ruby:

Figura 3 - Estrutura do if

```
iniciando.rb
1  if 10 < 5
2    puts "Verdadeiro"
3  end
4
```

Fonte: Autores.

Linha 1: digite `if`, posteriormente digite a condição para caso a condição seja `true` (verdadeiro) o código abaixo seja executado e para sinalizar que a estrutura dessa condicional acabou utilize o `end`, como foi feito na linha 3.

2.2.2 Else

Figura 4 - Estrutura do else

```
iniciando.rb
1  if 10 < 5
2    puts "Verdadeiro"
3  else
4    puts "primeira condição falsa"
5  end
```

Fonte: Autores.

Caso precise utilizar uma segunda condição se pode utilizar o `else`, como se usa em C, java e javascript.

3. Conclusão

Por fim, vimos alguns exemplos simples do uso do ruby, vendo um pouco da sua sintaxe, mostrando a sua forma simplificada de escrita com foco no aprendizado mais rápido e fácil entendimento.

4. Referências

“Sobre o Ruby”, <https://www.ruby-lang.org/pt/about/>

“Ruby Básico”, <https://www.caelum.com.br/apostila-ruby-on-rails/ruby-basico/>