

Guia de estilo para programar em Ruby

Rodrigo L. M. Flores
`mail@rodrigoflores.org`

Projeto Archlinux-BR

22 de agosto de 2009

Coisas a serem consideradas em um Guia de Estilo

- Guias de estilo são como um buffet e não como uma escritura sagrada
- Mas segui-los como um escritura sagrada faz nosso código ser melhor entendido pelos outros...
- Guias de estilo ajudam a manter um código mais bonito, mas não são a única coisa que mantém um código bonito
- Mesmo que um guia de estilo não seja seguido a risca, todo o código deve seguir as mesmas diretrizes mesmo guia

Organização do código

- 1 Cabeçalho com nome do autor e descrição do programa
- 2 Require's
- 3 Include's
- 4 Definições de classes e módulos
- 5 Programa principal
- 6 Código de teste

Sugestão de cabeçalho

Sugestão de cabeçalho ?

```
=begin
  * Name:
  * Description
  * Author:
  * Date:
  * License:
=end
```

Require x Include

Require faz o que o “include” de outras linguagens faz: roda o outro arquivo

Include pega todos os métodos de outro módulo e os inclui no módulo atual

Warnings

Warnings em Ruby

- Flag -w no interpretador (colocar no shebang também)

Variáveis globais

Quando usar ?

Para evitar problemas com conflitos de namespaces e referências acidentais

- One-liners
- Variáveis globais pré-definidas (\$. \$! \$?)

IMHO...

Cabe a cada um utilizar quando achar necessário. Mas tomar cuidado quando usar.

Nome de coisas

Classes CamelCase, substantivos

Módulos CamelCase, não é fácil recomendar

Constantes SCREAMING_SNAKE_CASE, substantivos

Siglas SCREAMINGCASE Exemplo: HTTP XML IME,

Nomes de métodos snake_case, se possível deve ser um verbo

Variáveis snake_case, substantivos (se a variável for local, pode abreviar)

Inicialização de (Hash—Array—String)

Dois jeitos possíveis

Escolher (e manter) um deles: `$1.new` ou `{}` `[]` `""`

Métodos booleanos

Devem terminar em “?”

Exemplos:

`Math.positive?`

`Stack.empty?`

Ponto e vírgula

Não usar isso!

Exemplos:

```
($stderr.puts "Error!"; exit 1) unless x == 2
```

Associações múltiplas

Use, mas tome cuidado com referências

Exemplo de pegadinha :

```
irb(main):001:0> foo = bar = Array.new  
=> []  
irb(main):002:0> foo  
=> []  
irb(main):003:0> bar  
=> []  
irb(main):004:0> foo << "baz"  
=> ["baz"]  
irb(main):005:0> foo  
=> ["baz"]  
irb(main):006:0> bar  
=> ["baz"]
```

Tamanho da linha

Máximo

80 ou 79.

E se não couber em uma linha o que eu preciso fazer ?

Se o interpretador achar que uma linha não faz sentido, ele tentará achar uma continuação na linha de baixo.

Recuo (aka i(n?)dentação)

2, 4, 8 ? Espaços, tabulações?

2 espaços. Mas isso não é uma consenso!

Linhas em branco

Onde quantas linhas em branco eu deixo?

- Deixar 2 linhas em branco entre cada definição de classe e módulo
- Deixar 1 linha em branco entre cada definição de método

Espaços em branco

Quando usar ?

Atribuição:

`a = 1` e não `a=1`

Quando não usar ?

Valores padrões em argumentos:

`def foo(a, b=1)` e não `def foo(a, b = 1)`

Não separar lista de argumentos da chamada do método:

`foo.bar(a)` e não `foo.bar (a)`

“Ou igual”

`|| =`

Use e abuse!

`x ||= 2`

Mesma coisa que:

`x = 2 if x.nil?`

Sites úteis

- Guia de estilo (mais prolixo) :
`http://www.caliban.org/ruby/rubyguide.shtml`
- Guia de estilo (mais resumido) : `http://github.com/chneukirchen/styleguide/tree/master`

Dúvidas

Contato :

E-mail flores@archlinux-br.org

XMPP <im@rodrigoflores.org>

Site <http://rodrigoflores.org>

Site do arch-br <http://flores.archlinux-br.org>

Blog <http://blog.rodrigoflores.org>

Twitter [rodrigoflores](#)

Identi.ca [rodrigoflores](#)

Jaiku [flores](#)