



RUBY



Características

- Linguagem interpretada.
- Multiparadigma.
- Tipagem forte e dinâmica.
- Gerenciamento automático de memória.
- Multiplataforma.
- Open-source.

Métodos "attribute accessors"

- Ruby:

- class Bichos
 - attr_accessor :animais
- end
- exemplo = Bichos.new()
- exemplo.animais = ["cachorro", "gato", "cavalo"]
- print exemplo.animais

- Java:

- import java.util.*;
- public class Bichos{
 - Vector animais = new Vector();
 - public String getAnimais(){
 - return this.animais.toString();
 - }
 - public void setAnimais(String animal){
 - this.animais.add(animal);
 - }
- }

Métodos que retornam true ou false terminam com interrogação

- Ruby:
- `animais = ["cachorro", "gato", "cavalo"]`
- `print animais.include? "gato"`

Métodos que alteram o objeto são terminados com ponto de exclamação

- Ruby:
- `animais = ["cachorro", "gato", "cavalo"]`
- `animais.slice!(1)`
- `print animais.include? "gato"`

Métodos Slice! e Include? (java)

- Java:

- `import java.util.*;`
- `public class Main{`
- `public static void main(String[] args) {`
- `List animais = new ArrayList();`
- `boolean achou = false;`
- `animais.add("cachorro");`
- `animais.add("gato");`
- `animais.add("cavalo");`
- `//===== slice! =====`
- `animais.remove(1);`
- `//=====`

- `//===== include? =====`
- `for(int i = 0; i<array.size(); i++){`
- `if(animais.get(i).equals("gato")){`
- `achou = true;`
- `break;`
- `}`
- `}`
- `//=====`
- `System.out.println(achou);`
- `}`
- `}`

Tudo é objeto

- Ruby:

- `def pares(limite)`
- `1.upto(limite) {`
- `|i| puts i if i % 2 == 0`
- `}`
- `end`

- Java:

- `public void pares(int limite){`
- `for(int i = 1; i<= limite; i++){`
- `if(i % 2)System.out.println(i);`
- `}`
- `}`

Monkey Patch

```
| class Fixnum  
|   alias :soma_velha :+  
|   def +(valor)  
|     return self.to_s + valor if valor.is_a? String  
|     soma_velha(valor)  
|   end  
| end  
| print 1 + "2"
```


Bibliografia

Wikipedia - Ruby(linguagem de programação): [https://pt.wikipedia.org/wiki/Ruby_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Ruby_(linguagem_de_programa%C3%A7%C3%A3o))

Sitepoint - The History of Ruby: <https://www.sitepoint.com/history-ruby/>

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

DEVMEDIA - Ruby on Rails Tutorial: <https://www.devmedia.com.br/ruby-on-rails-tutorial/31285>

DEVMEDIA - Conhecendo a Linguagem Ruby: <https://www.devmedia.com.br/conhecendo-a-linguagem-ruby/8226>

EU SOU DEV - Dúvida cruel: Ruby ou PHP?: <https://eusoudev.com.br/ruby-ou-php/>

Caelum - A linguagem Ruby: <https://www.caelum.com.br/apostila-ruby-on-rails/a-linguagem-ruby/#2-1-a-historia-do-ruby-e-suas-caracteristicas>

Arthur de Moura Del Esposte - Orientação a Objetos com Ruby: <https://www.ime.usp.br/~esposte/documents/aula-ruby/aula01/Aula01.pdf>

Carlos Henrique Costa Junior, Daniel de Andrade, Darlan Pedro de Campos e Guilherme Silvestre Gomes - Ruby:
<https://www.inf.ufsc.br/~frank.siqueira/INE5612/Seminario2010.1/Ruby.pdf>

AkitaOnRails - Micro-Tutorial de Ruby - Parte I :<https://www.akitaonrails.com/2008/11/10/micro-tutorial-de-ruby-parte-i>