Ruby master - Feature #6243
Make ERB Ruby's Super Power
04/02/2012 06:31 AM - trans (Thomas Sawyer)

Status: Rejected
Priority: Normal
Assignee:
Target version: 2.0.0

Description
As a Super Ruby Developer
I want to make Ruby Super Developery
Given wimpy Ruby, coding is arduous:

class Letters
def a
  "Aa"
end
def b
  "Bb"
end

# Oh, dear St. V! It's just too long. I don't want to go on!

Then Super Ruby saves the day!

class Letters
  <% (a..z).each do |letter| %>
  def <%= letter %>
    "<%= letter.upcase %>\n<%= letter %>
  end
  <% end %>
end

letters = Letters.new

letters.a #=> "Aa"
letters.b #=> "Bb"

Imagine the power of Ruby ** Ruby! It's like putting Ruby in a phone booth and out pops Ruby with big on chest. And like that, Super Ruby destroys all super villain languages like PHP!

And you can't reject b/c this issue is all Gherkin which makes it legit.

History
#1 - 04/03/2012 12:40 AM - tenderlovemaking (Aaron Patterson)
- Category changed from core to Joke

#2 - 04/03/2012 04:43 AM - kernigh (George Koehler)
=begin
The original poster's code (for ERB) fails with SyntaxError.

This code (for plain Ruby) works:
class Letters
  (a..z).each do |letter|
    result = letter.upcase + letter
    define_method(letter) { result }
  end
=end

#3 - 04/03/2012 05:12 AM - trans (Thomas Sawyer)
You actually tried it!? Sweet!

Looking it over the error was probably b/c the first `<%=` should not have the `=`.

Of course this was for April Fool's Day. As your code shows Ruby is quite capable of meta-programming without ERB based macro system... albeit not quite as "super"! :)

I note, though, it's not quite useless either. I've used similar idea to update VERSION = in version.rb file of some projects before.

#4 - 04/04/2012 11:40 AM - tenderlovemaking (Aaron Patterson)
- Status changed from Open to Rejected

Closing since April 1st has completed. :-}