

Ruby trunk - Feature #749

String#indent, String#tabto, etc.

11/13/2008 03:01 AM - trans (Thomas Sawyer)

Status:	Rejected
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	2.0.0
Description	
<p>=begin There are few indention related String extensions I find myself frequently reusing. Ruby really has nothing built-in that addresses this, so I'd like to the following methods, or something like them, ultimately make it into Ruby core.</p> <pre># Preserves relative tabbing. # The first non-empty line ends up with n spaces before nonspace. def tabto(n) if self =~ /\S/ indent(n - \$1.length) else self end end # Indent left or right by n spaces. # (This used to be called #tab and aliased as #indent.) def indent(n) if n >= 0 gsub(/'/, '' * n) else gsub(/^{0,#{-n}}/, "") end end # Aligns each line n spaces. def tab(n) gsub(/^\s*/, '' * n) end # Provides a margin controlled string. # # x = %Q{ # This # is # margin controlled! # } # def margin(n=0) d = ((/\A.\n\s(.)/.match(self)) (/^\A\s*(.)/.match(self)))[1] return " unless d if n == 0 gsub(/\n\s*\Z/, "").gsub(/\s*#{d}/, "") else gsub(/\n\s*\Z/, "").gsub(/\s*#{d}/, '' * n) end end end</pre>	
<p>These methods come in very handy in avoiding the need to shove HERE docs up against the left margin, or use repetitive string concatenations. I've seen others mention them from time to time as well, so I know I'm not the only one using them.</p> <p>Of course, they'd need to be converted to C, but given the size and character of the definitions, I imagine it would be a fairly straight forward translation.</p>	

(Giving credit where it is due: The first three methods were originally written by Gavin Sinclair. The later method, if I recall correctly, by Peter Vanbroekhoven).
=end

History

#1 - 11/29/2008 04:38 PM - ko1 (Koichi Sasada)

- Assignee set to matz (Yukihiko Matsumoto)

=begin

=end

#2 - 09/14/2010 04:49 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

=begin

=end

#3 - 02/13/2012 08:34 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Rejected

I'm rejecting this feature ticket because no progress has been made for a long time. See [ruby-42391].

Sorry, I can say few comments. The use case is trivial for me.

There is nothing wrong with the proposal, I think. But matz did not answer, and will not, just keeping this ticket open.

I suggest you, or those who want these features, to get a fresh start; create a new ticket (with a patch if possible), and appeal to matz.

--

Yusuke Endoh mame@tsg.ne.jp