

Ruby - Feature #4632

Regexp instance method | for creating Regexp.unions

04/29/2011 09:59 PM - rbjl (Jan Lelis)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description =begin Regexp.union is a great feature, but it is rarely used. I can imagine, the following syntax would be quite popular: For example: /Ruby\d/ /test/i "cheat" #=> Regexp.union(Regexp.union(/Ruby\d/, /test/i), "cheat") It would also be cool as String method, so one can do: "jruby" "rbx" #=> /jruby rbx/ Example implementation: class Regexp def (arg) Regexp.union self, arg.is_a?(Regexp) ? arg : arg.to_s end end class String def (arg) Regexp.union self, arg.is_a?(Regexp) ? arg : arg.to_s end end =end		

History

#1 - 04/29/2011 10:23 PM - rkh (Konstantin Haase)

=begin
I like it for Regexp, but a string method would violate the POLS, imho.

Konstantin

On Apr 29, 2011, at 14:59 , Jan Lelis wrote:

Issue [#4632](#) has been reported by Jan Lelis.

Feature [#4632](#): Regexp instance method | for creating Regexp.unions
<http://redmine.ruby-lang.org/issues/4632>

Author: Jan Lelis
Status: Open
Priority: Normal
Assignee:
Category:
Target version:

Regexp.union is a great feature, but it is rarely used. I can imagine, the following syntax would be quite popular:

For example:
/Ruby\d/ | /test/i | "cheat" #
=end

#2 - 04/29/2011 10:33 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Rejected

=begin

We once had this feature, `eregex.rb`, but dropped since 1.9. No one nursed that lib.

I don't stop you to have such tiny lib as a gem. If that gem was used widely, chances are that we might invite that as our standard again someday.

=end

#3 - 04/29/2011 11:22 PM - rbjl (Jan Leis)

=begin

Thanks for the quick answer.

=end