

Ruby - Feature #13212

Syntax proposal: don't require begin-end to rescue exceptions inside do-end blocks

02/13/2017 09:02 PM - Nowaker (Damian Nowak)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description Blocks are ubiquitous in Ruby world. There's as many blocks in every Ruby code as methods - if not even more. It would be very nice to treat blocks as first class citizens like methods and reduce the undesirable begin-end boilerplate inside blocks for exception handling. Here's the syntax proposal: <pre>arr.each do e do_something(e) rescue MyException => e # handle exception end</pre> Just like in methods: <pre>def method do_something rescue MyException => ex # handle exception end</pre> Currently begin-end block is required for handling exceptions inside blocks: <pre>arr.each do e begin do_something(e) rescue MyException => ex # handle exception end end</pre> Whether the same syntax would apply for {} block/lambda syntax is debatable. {} are typically one-liners and I barely ever saw a begin-rescue-end inside them. Thank you -Damian Nowak		
Related issues:		
Has duplicate Ruby - Feature #12906: do/end blocks work with ensure/rescue/else		Closed

History

#1 - 02/14/2017 06:50 AM - hsb (Hiroshi SHIBATA)

- Has duplicate Feature #12906: do/end blocks work with ensure/rescue/else added

#2 - 02/14/2017 06:51 AM - hsb (Hiroshi SHIBATA)

- Status changed from Open to Closed

It's accepted by <https://bugs.ruby-lang.org/issues/12906>

You can use this syntax after Ruby 2.5.