

Ruby master - Feature #17097

`map_min`, `map_max`

07/31/2020 12:33 PM - sawa (Tsuyoshi Sawada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>min, min_by, max, max_by return the element that leads to the minimum or the maximum value, but I think it is as, or even more, frequent that we are interested in the minimum or the maximum value itself rather than the element. For example, to get the length of the longest string in an array, we do:</p> <pre>%w[aa b cccc dd].max_by(&:length).length # => 4 %w[aa b cccc dd].map(&:length).max # => 4</pre> <p>I propose to have methods that return the minimum or the maximum value. Temporarily calling them map_min, map_max, they should work like this:</p> <pre>%w[aa b cccc dd].map_max(&:length) # => 4</pre> <p>map_min, map_max are implementation-centered names, so perhaps better names should replace them, just like yield_self was replaced by then.</p>	

History

#1 - 07/31/2020 12:37 PM - nobu (Nobuyoshi Nakada)

Then we'll need map_ versions for all Enumerable methods.

#2 - 07/31/2020 12:41 PM - sawa (Tsuyoshi Sawada)

- Description updated

#3 - 07/31/2020 12:43 PM - sawa (Tsuyoshi Sawada)

- Description updated

#4 - 07/31/2020 01:51 PM - greggzst (Grzegorz Jakubiak)

nobu (Nobuyoshi Nakada) wrote in [#note-1](#):

Then we'll need map_ versions for all Enumerable methods.

Exactly, I don't see any good use case in that apart from being lazy and just using one method call. It seems to me that this kind of proposal is too much. I mean there are more pressing issues or features of the language that the team has to focus on instead of dealing with proposals like map_tally or whatever.

#5 - 07/31/2020 06:03 PM - Eregon (Benoit Daloze)

I actually regularly wanted such functionality, but min_by/max_by do not return the result of the block but the element (which makes sense). And .map {}.min/max works but is inefficient by generating an intermediate Array.

Maybe we should add some keyword argument to min_by/max_by?

#6 - 07/31/2020 06:06 PM - Eregon (Benoit Daloze)

To put in context, consider that the expression might be much longer than .length. Then repeating it is not elegant and is duplicated code.

#7 - 08/01/2020 12:56 AM - marcandre (Marc-Andre Lafortune)

[nobu \(Nobuyoshi Nakada\)](#) is right, we're not going to add map_ for everything.

Eregon (Benoit Daloze) wrote in [#note-6](#):

Then repeating it is not elegant and is duplicated code.

I don't see why there would be repetition.

Just do `enum.map { very_long_expression }.max...`

Please benchmark the time it takes to generate the intermediate array. My guess is that it's typically negligible.

#8 - 08/01/2020 03:53 AM - sawa (Tsuyoshi Sawada)

I do not understand why the proposal has to be extended to all other Enumerable methods.

My point is semantic. I do not see that there are significantly more use cases where I am interested in the element that is related to the min/max value but am not interested in the min/max value than the use cases of the converse.