Ruby master - Feature #15123
Enumerable#compact proposal
09/14/2018 05:30 PM - printercu (Max Melentiev)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Hi!

While Enumerable does not provide #compact method, it requires changing code in some cases to substitute array with enumerator. For example, to reduce memory usage it's usual to change large_array.map { to_heavy_object }.chained_methods to large_array.lazy.... However if chained_methods contains compact, this change will fail. Replacing compact with reject(&:nil?) fixes it. What do you think about adding #compact to Enumerable?

**History**

#1 - 09/14/2018 07:43 PM - greggzst (Grzegorz Jakubiak)
I'm in favor of this proposal. It simplifies working with large and small collections so one doesn't have to remember that can't use #compact when enumerator is returned and have to fall back to #reject(&:nil?).

#2 - 09/15/2018 02:37 AM - marcandre (Marc-Andre Lafortune)
- Assignee set to matz (Yukihiro Matsumoto)

Proposal seems acceptable to me.

Just to be clear: I imagine that Lazy#compact would still be lazy. Also compact is roughly select(&:itself), not reject(&:nil?) which would wrongly keep false.

#3 - 09/15/2018 07:45 AM - Eregon (Benoit Daloze)
marcandre (Marc-Andre Lafortune) wrote:

 Also compact is roughly select(&:itself), not reject(&:nil?) which would wrongly keep false.

No, #compact only removes nil: ["a",false,nil].compact => ["a",false].

#4 - 09/15/2018 11:06 AM - shevegen (Robert A. Heiler)
I think if the meaning is consistent (e. g. .compact meaning to eliminate nil values from a given collection) then this seems ok. Perhaps this could be added for the next developer meeting.

#5 - 09/15/2018 02:38 PM - marcandre (Marc-Andre Lafortune)
Eregon (Benoit Daloze) wrote:

 No, #compact only removes nil: ["a",false,nil].compact => ["a",false].

Lol, ouch, not sure how I could be so confused when I wrote that, sorry!

#6 - 09/24/2018 11:07 PM - nobu (Nobuyoshi Nakada)
- Backport deleted (2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN)
- Tracker changed from Bug to Feature

#7 - 05/20/2019 07:23 AM - shyouhei (Shyouhei Urabe)
Just leaving a comment that I wanted this method in my pet project just now. So +1.
I don't see enough demand for compact where we have reject(&:nil?). Any additional use-case?

Matz.

- Status changed from Open to Feedback

Its presence in Array and Hash make it more of a common interface that I could see being defined for Enumerable in general, though the immediate usecases are around Lazy. As mentioned above, sometimes one wants to use Enumerators directly or returns one from an Enumerable method which can cause some conflicts of available methods.

I believe it could be considered surprising that compact does not necessarily work with all collection-like types.