Ruby trunk - Feature #15323

[PATCH] Proposal: Add Enumerable#filter_map

11/20/2018 11:59 AM - alfonsojimenez (Alfonso Jiménez)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

This is a proposal for a combined filter + map method (https://bugs.ruby-lang.org/issues/5663).

This method both filters and maps the elements of an enumerable in just one iteration:

```
(1..10).filter_map { |i| i * 2 if i.even? } #=> [4, 8, 12, 16, 20]
```

GitHub PR: https://github.com/ruby/ruby/pull/2017

Related issues:

Related to Ruby trunk - Feature #5663: Combined map/select method	**Assigned**

Associated revisions

Revision 0acbdd1e - 05/23/2019 05:39 AM - alfonsojimenez (Alfonso Jiménez)

Adding Enumerable#filter_map

[Feature #15323]
Closes: https://github.com/ruby/ruby/pull/2017

History

#1 - 11/20/2018 12:01 PM - alfonsojimenez (Alfonso Jiménez)

- Description updated

#2 - 11/20/2018 02:23 PM - alfonsojimenez (Alfonso Jiménez)

- File deleted (0001-Adding-Enumerable-filter_map.patch)

#3 - 11/20/2018 02:24 PM - alfonsojimenez (Alfonso Jiménez)

- File 0001-Adding-Enumerable-filter_map.patch added

#4 - 11/20/2018 02:29 PM - tny (Tony Sunny)

Could't we use reduce for this?

```
(1..10).reduce([]) { |a, i| i.even? ? a << (i * 2) : a }
```

#5 - 11/20/2018 05:07 PM - shevegen (Robert A. Heiler)

I think the functionality, that is to combine .filter (be it select
or reject, is secondary to me), and .map, could be useful. I don't
really need it myself but I find it is not entirely out of the question
that others may find it useful.

There is, IMO, only one real drawback, if we ignore the functionality
aspect (where you'd have to ask matz anyway), and this is that I
think the two-word methods can be quite clumsy.

Not just .filter_map but also .yield_self, which eventually had an
alias called .then. If we ignore the question as to whether .then
is a good name (or .yield_self), one advantage that .then has
is that it is shorter.

Succinct expression is not always necessarily the best; but in this
case, I think single-word methods are very often better than two-word
methods.

.reduce() has, in my opinion, a slight other disadvantage, and that is that people have to explicitely pass an , which is not always easy to remember. (For me it is hard to remember because I rarely use .reduce either).

This is just my opinion, though. I do not really have any strong pro or con way about the feature itself; only a very tiny dislike of .filter_map as name. But it is not really a strong contra opinion either way. (My biggest look ahead is on ruby's jit/mjit ... :D)

#6 - 12/14/2018 07:58 PM - devpolish (Nardo Nykolyszyn)

(1..10).map { |e| e.even? ? (e * 2) : e }

#7 - 12/14/2018 08:32 PM - oleynikov (Alexander Oleynikov)

nardonykolyszyn@gmail.com wrote:

> (1..10).map { |e| e.even? ? (e * 2) : e }

Yeah, but without #filter this is still an array with 10 elements.

#8 - 12/17/2018 03:43 AM - phluid61 (Matthew Kerwin)

tny (Tony Sunny) wrote:

> Could't we use reduce for this?
>
> ```
> (1..10).reduce([]) { |a, i| i.even? ? a << (i * 2) : a }
> ```

Yep, that's mentioned in the original ticket too. There's also #each_with_object that lets you write the block almost the same as in the proposal:

```
(1..10).each_with_object([]) { |i, a| a << i * 2 if i.even? }
```

The big difference here is you can capture nil/false values, because the filter test is explicitly separated from the map operation.

#9 - 02/13/2019 02:35 AM - shugo (Shugo Maeda)

- Related to Feature #5663: Combined map/select method added

#10 - 02/13/2019 02:41 AM - shugo (Shugo Maeda)

+1 for filter_map.

Matz agreed the feature itself before: https://bugs.ruby-lang.org/issues/5663#note-42
The name filter_map is good because other languages have similar names (e.g., filter-map in Scheme).

#11 - 04/25/2019 09:48 AM - alfonsojimenez (Alfonso Jiménez)

- File deleted (0001-Adding-Enumerable-filter_map.patch)

#12 - 04/25/2019 09:49 AM - alfonsojimenez (Alfonso Jiménez)

- File 0001-Adding-Enumerable-filter_map.patch added

#13 - 04/25/2019 09:54 AM - alfonsojimenez (Alfonso Jiménez)

I've updated the patch file increasing the ruby version in *spec/ruby/core/enumerable/filter_map_spec.rb*

Enumerable#filter_map was already accepted in the last developers meeting:
https://docs.google.com/document/u/2/d/e/2PACX-1vTUCmj7aUdnMAdunG0AZo0AdWK-9jvfXcB7DWYmzGtmPc0IuIPGn7eLARoR5tBd6XUUB08W-hH74k-T/pub

#14 - 04/30/2019 09:35 PM - greggzst (Grzegorz Jakubiak)

alfonsojimenez (Alfonso Jiménez) wrote:

> I've updated the patch file increasing the ruby version in *spec/ruby/core/enumerable/filter_map_spec.rb*
>
> Enumerable#filter_map was already accepted in the last developers meeting:
> https://docs.google.com/document/u/2/d/e/2PACX-1vTUCmj7aUdnMAdunG0AZo0AdWK-9jvfXcB7DWYmzGtmPc0IuIPGn7eLARoR5tBd6XUUB08W-hH74k-T/pub

Does the syntax allow for this kind of code?

```
(1..10).filter_map(&:even?) { |i| i * 2 }
```

#15 - 05/22/2019 06:06 AM - matz (Yukihiro Matsumoto)

I accepted the proposal at the last developer meeting but forgot to post here.
I do reject having both block and block argument at the same time. [ruby-core:92505]
Regarding filter_map!, submit a new proposal, if you really needed (with the use-case).

Matz.

#16 - 05/23/2019 05:51 AM - alfonsojimenez (Alfonso Jiménez)

- Status changed from Open to Closed

Applied in changeset git|0acbdd1ed0d2302743525a5188cc5a0d6251680c.

Adding Enumerable#filter_map

[Feature #15323]
Closes: https://github.com/ruby/ruby/pull/2017

#17 - 05/23/2019 03:46 PM - nobu (Nobuyoshi Nakada)

IIRC, at the last meeting (20190522), the conclusion was that this method should select non-nil values only, like as Array#compact.
Am I correct?

Files

0001-Adding-Enumerable-filter_map.patch	4.61 KB	04/25/2019	alfonsojimenez (Alfonso Jiménez)