

Ruby master - Feature #15143

Extend `Enumerable#to_h`

09/20/2018 02:23 AM - sawa (Tsuyoshi Sawada)

Status:	Closed
Priority:	Normal
Assignee:	nobu (Nobuyoshi Nakada)
Target version:	
Description	
Often, we call <code>Array#to_h</code> to the result of <code>Enumerable#map</code> :	
<pre>(1..5).map{ x [x, x ** 2]}.to_h #=> {1=>1, 2=>4, 3=>9, 4=>16, 5=>25}</pre>	
I am thinking of a feature to do this in a single method call.	
Currently, <code>Enumerable#to_h</code> does not accept a block. I propose that, when <code>Enumerable#to_h</code> is called with a block (that has a subarray representing a key-value pair), return a hash that would be returned by applying the block to <code>map</code> , and <code>to_h</code> to the result:	
<pre>(1..5).to_h{ x [x, x ** 2]} #=> {1=>1, 2=>4, 3=>9, 4=>16, 5=>25}</pre>	
Ideally, I request this to be done internally to Ruby without creating an intermediate parent array.	
Related issues:	
Is duplicate of Ruby master - Feature #10208: Passing block to <code>Enumerable#to_h</code>	Closed

Associated revisions

Revision [abe75149](#) - 09/20/2018 03:06 PM - nobu (Nobuyoshi Nakada)

`Enumerable#to_h` with block and so on

[Feature #15143]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64794 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision [64794](#) - 09/20/2018 03:06 PM - nobu (Nobuyoshi Nakada)

`Enumerable#to_h` with block and so on

[Feature #15143]

Revision [64794](#) - 09/20/2018 03:06 PM - nobu (Nobuyoshi Nakada)

`Enumerable#to_h` with block and so on

[Feature #15143]

History

#1 - 09/20/2018 02:27 AM - sawa (Tsuyoshi Sawada)

I forgot to mention that people alternatively do:

```
(1..5).each_with_object({}){|x, h| h[x] = x ** 2}
#=> {1=>1, 2=>4, 3=>9, 4=>16, 5=>25}
```

which may be more optimal than calling `map`. The proposed feature also has the effect of making this technique unnecessary (although the proposal is not as effective as this in that the former creates intermediate subarrays).

#2 - 09/20/2018 02:41 AM - matz (Yukihiro Matsumoto)

- Assignee set to nobu (Nobuyoshi Nakada)

Sounds reasonable. Accepted.

Matz.

#3 - 09/20/2018 02:47 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #10208: Passing block to Enumerable#to_h added

#4 - 09/20/2018 02:47 AM - nobu (Nobuyoshi Nakada)

- Subject changed from Extend `Enumerrable#to_h` to Extend `Enumerable#to_h`

Have you changed the mind since [#10208](#)?

#5 - 09/20/2018 03:04 AM - mame (Yusuke Endoh)

If the block returns an array whose length is > 2, how does it behave?

```
%w(aa1 ab2 ba3 bb4).to_h {|s| s.chars }  
#=> { "a" => { "a" => "1",  
              "b" => "2" },  
      "b" => { "a" => "3",  
              "b" => "4" } }
```

I expect very useful trie generator :-)

#6 - 09/20/2018 06:25 AM - marcandre (Marc-Andre Lafortune)

nobu (Nobuyoshi Nakada) wrote:

Have you changed the mind since [#10208](#)?

Or since... [#666](#)!

#7 - 09/20/2018 06:54 AM - sawa (Tsuyoshi Sawada)

Sorry that I missed the previous duplicate proposals (I found that I had even commented on one of them, which I had forgotten about). But it looks like four years have passed since then, and the situation has changed perhaps. Or, it may have worth proposing again to indicate that many people want such feature.

I am grateful to Matz for accepting this.

#8 - 09/20/2018 07:14 AM - sawa (Tsuyoshi Sawada)

mame (Yusuke Endoh) wrote:

If the block returns an array whose length is > 2, how does it behave?

```
%w(aa1 ab2 ba3 bb4).to_h {|s| s.chars }  
#=> { "a" => { "a" => "1",  
              "b" => "2" },  
      "b" => { "a" => "3",  
              "b" => "4" } }
```

I expect very useful trie generator :-)

My original proposal is to follow the behaviour of the current map({...}).to_h, which means to raise ArgumentError: wrong array length at 0 (expected 2, was 3).

However, mame's proposal to extend the feature also sounds interesting and good. Though, after having Matz' acceptance of this proposal, I am not sure whether we can still get another approval on top of it from him assuming mame's extension, or whether the extension should be another issue.

#9 - 09/20/2018 07:26 AM - matz (Yukihiro Matsumoto)

I changed my mind when I approved to_h.

Matz.

#10 - 09/20/2018 07:27 AM - matz (Yukihiro Matsumoto)

I don't agree with [mame \(Yusuke Endoh\)](#)'s idea. The return value from the block should be strictly 2 elements array.

Matz.

#11 - 09/20/2018 07:32 AM - sawa (Tsuyoshi Sawada)

Matz, thanks for confirmation. I am happy with my original proposal.

#12 - 09/20/2018 11:48 AM - nobu (Nobuyoshi Nakada)

OK, not only Enumerable, but Array, Hash, ENV, NilClass and Struct too?

#13 - 09/20/2018 12:38 PM - nobu (Nobuyoshi Nakada)

nil isn't an Enumerable, so it's irrelevant.

#14 - 09/20/2018 03:06 PM - nobu (Nobuyoshi Nakada)

- *Status changed from Open to Closed*

Applied in changeset [trunk|r64794](#).

Enumerable#to_h with block and so on

[Feature [#15143](#)]

#15 - 09/21/2018 02:23 AM - sawa (Tsuyoshi Sawada)

nobu (Nobuyoshi Nakada) wrote:

OK, not only Enumerable, but Array, Hash, ENV, NilClass and Struct too?

Thanks for reminding me about that and implementing it.