

Ruby master - Feature #11797

`Enumerator#with_object` with multiple objects

12/09/2015 04:15 PM - sawa (Tsuyoshi Sawada)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>Sometimes, when working with <code>Enumerator#with_object</code>, I want to keep some additional temporary objects besides the one to return. A use case is as follows (I got this from this StackOverflow question: http://stackoverflow.com/questions/3418123). Suppose I have an enumerator created from an array:</p> <pre>e = ["a", "b", "c", "c", "a", "c"].to_enum</pre> <p>and want to get an array of its repeated elements in the order they are repeated (i.e., appears for the second time):</p> <pre># => ["c", "a"]</pre> <p>I can do it using <code>Enumerator#with_object</code> like this:</p> <pre>e.to_enum.with_object([{}], []){ c, (h, a) h[c] ? a.push(c) : h.store(c, true)}.last.uniq</pre> <p>Here, I am getting the array referred to as <code>a</code> in the block, but besides that, I am using a temporal hash <code>h</code>. I thought it would be nice if <code>Enumerator#with_object</code> accepts one or more objects, pass them individually as block arguments, and returns only the last argument so that I can do this:</p> <pre>e.to_enum.with_object({}, []){ c, h, a h[c] ? a.push(c) : h.store(c, true)}.uniq</pre>	

History

#1 - 12/09/2015 04:17 PM - sawa (Tsuyoshi Sawada)

The StackOverflow question link was wrong. It should be: <http://stackoverflow.com/questions/34181231>.

#2 - 12/18/2015 04:38 AM - shugo (Shugo Maeda)

Hello,

I've removed his address from ruby-core and ruby-talk.

2015-12-18 13:27 GMT+09:00 Joseph Jones joeyi5216@gmail.com:

Joseph Jones liked your message with Boxer.

On December 9, 2015 at 09:15:54 MST, sawadatsuyoshi@gmail.com wrote:

Issue [#11797](#) has been reported by Tsuyoshi Sawada.

Feature [#11797](#): `Enumerator#with_object` with multiple objects
<https://bugs.ruby-lang.org/issues/11797>

- Author: Tsuyoshi Sawada
- Status: Open
- Priority: Normal

* Assignee:

Sometimes, when working with `Enumerator#with_object`, I want to keep some additional temporary objects besides the one to return. A use case is as follows (I got this from this StackOverflow question: <http://stackoverflow.com/questions/3418123>). Suppose I have an enumerator created from an array:

```
e = ["a", "b", "c", "c", "a", "c"].to_enum
```

and want to get an array of its repeated elements in the order they are repeated (i.e., appears for the second time):

=> ["c", "a"]

I can do it using Enumerator#with_object like this:

```
e.to_enum.with_object({}, []){|c, (h, a)| h[c] ? a.push(c) : h.store(c, true)}.last.uniq
```

Here, I am getting the array referred to as a in the block, but besides that, I am using a temporal hash h. I thought it would be nice if Enumerator#with_object accepts one or more objects, pass them individually as block arguments, and returns only the last argument so that I can do this:

```
e.to_enum.with_object({}, []){|c, h, a| h[c] ? a.push(c) : h.store(c, true)}.uniq
```

--

<https://bugs.ruby-lang.org/>

--

Shugo Maeda

#3 - 05/09/2019 06:21 AM - sawa (Tsuyoshi Sawada)

I now realize that, for the given use case, I can use with_object multiple times as follows:

```
e
.each.with_object({}).with_object([]) {|(c, h), a| h[c] ? a.push(c) : h.store(c, true)}
.uniq
```

So I withdraw this feature request. Please close it.

#4 - 05/09/2019 06:49 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed