Ruby master - Feature #9071
Enumerable#slice_after
11/02/2013 01:21 AM - sawa (Tsuyoshi Sawada)

Status: Closed
Priority: Normal
Assignee: akr (Akira Tanaka)
Target version: 

Description
I see in this discussion: http://ruby.11.x6.nabble.com/ruby-dev-38392-Enumerable-gather-each-td3534746.html that Enumerable#slice_before was named as such, having in mind the possibility of Enumerable#slice_after being implemented in the future. I feel the former convenient, but believe the latter should be as well, and am wondering why the latter was not implemented at the same time. I request it to be implemented.

Associated revisions
Revision ddd15584 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.
  Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45981 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.
  Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.
  Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.
  Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.
  Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

08/28/2022
Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.

Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.

Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Revision 45981 - 05/18/2014 12:06 AM - akr (Akira Tanaka)

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.

Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

History
#1 - 11/02/2013 12:23 PM - akr (Akira Tanaka)

2013/11/2 sawa (Tsuyoshi Sawada) sawadatsuyoshi@gmail.com:

Feature #9071: Enumerable#slice_after
https://bugs.ruby-lang.org/issues/9071

I see in this discussion: http://ruby.11.x6.nabble.com/ruby-dev-38392-Enumerable-gather-each-td3534746.html that Enumerable#slice_before was named as such, having in mind the possibility of Enumerable#slice_after being implemented in the future. I feel the former convenient, but believe the latter should be as well, and am wondering why the latter was not implemented at the same time. I request it to be implemented.

The main reason is no one requested.
I have not enough time to implement it now.

If you'll implement Enumerable#slice_after, I recommend not implement state management
(the 3rd form of Enumerable#slice_before:
enum.slice_before(initial_state) { |elt, state| bool }).
Now, I think state management should be separated to another method as
http://bugs.ruby-lang.org/issues/8840#note-5

Tanaka Akira

#2 - 05/10/2014 11:35 AM - akr (Akira Tanaka)
- File slice_after.patch added
- Assignee set to akr (Akira Tanaka)
I implemented Enumerable#slice_after.

I found two requests for this feature:
This issue and
http://stackoverflow.com/questions/22622156/how-to-implement-slice-after-or-group-certain-elements-with-certain-subsequent

I wrote the document with an example follows:

```
% ./ruby -e '
lines = ["foo
", "bar\n", "baz
", 
", 
", "qux
"]
e = lines.slice_after(/(?<!\)\n\z/)
p e.to_a
p e.map {ll[0...-1].map {ll[1].sub(/\n\z/, ").join + ll.last }'
["foo
"], ["bar\n", "baz
"], ["\n"], ["qux
"]
"foo
", "barbaz
", "\n", "qux
"]
```

This concatenates continuation lines.

More useful (catchy) examples may be helpful to persuade matz.

Any idea?

#3 - 05/12/2014 09:47 AM - akr (Akira Tanaka)
- File slice_after2.patch added

I updated the patch to simplify argument handling.

#4 - 05/17/2014 06:33 AM - matz (Yukihiro Matsumoto)
Accepted.

Matz

#5 - 05/18/2014 12:06 AM - akr (Akira Tanaka)
- Status changed from Open to Closed
- % Done changed from 0 to 100

Applied in changeset r45981.

- enum.c: Enumerable#slice_after implemented.
- enumerator.c: Enumerator::Lazy#slice_after implemented.

Requested by Tsuyoshi Sawada. [ruby-core:58123] [Feature #9071]

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>slice_after.patch</td>
<td>7.01 KB</td>
<td>05/10/2014</td>
<td>akr (Akira Tanaka)</td>
</tr>
<tr>
<td>slice_after2.patch</td>
<td>6.87 KB</td>
<td>05/12/2014</td>
<td>akr (Akira Tanaka)</td>
</tr>
</tbody>
</table>