Ruby - Bug #4757

Attempt to make Enumerator docs more clear (patch included)

05/22/2011 04:28 AM - davetron5000 (David Copeland)

Status:	Closed	
Priority:	Normal	
Assignee:	drbrain (Eric Hodel)	
Target version:		
ruby -v:	-	**Backport:**

Description

The Enumerator docs aren't super clear; these make it more clear, at least to me, and also more consistent.

Associated revisions

Revision 6cfb4b61 - 05/22/2011 11:33 PM - drbrain (Eric Hodel)

- enumerator.c: Improve documentation. Patch by Dave Copeland.
  [Ruby 1.9 - Bug #4757]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@31704 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d137810d - 07/09/2011 12:25 AM - drbrain (Eric Hodel)

- enumerator.c: Remove "enumeration sequenced by".
  [Ruby 1.9 - Bug #4757]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32466 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 05/23/2011 08:33 AM - drbrain (Eric Hodel)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r31704.
David, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- enumerator.c: Improve documentation. Patch by Dave Copeland.
  [Ruby 1.9 - Bug #4757]

#2 - 05/25/2011 01:42 AM - mame (Yusuke Endoh)

- Status changed from Closed to Open

Hello,

- p e.next #=> 1
- p e.next #=> 2
- p e.next #=> 3
- p e.next #raises StopIteration
- puts e.next # => 1
- puts e.next # => 2
- puts e.next # => 3
- puts e.next # raises StopIteration

Why did you use puts than p?

- Note that enumeration sequence by next_values method does not affect other
- non-external enumeration methods, unless underlying iteration
- methods itself has side-effect, e.g. IO#each_line.
- Note that enumeration sequenced by +next_values+ does not affect other
- non-external enumeration methods, unless underlying iteration methods
- itself has side-effect, e.g. IO#each_line.

I cannot undestand what "enumeration sequence by next_values" is.
This is a problem of the original rdoc, not yours, though.
It might be good to remove "enumeration sequence by".

The old rdoc was written by akr. Akr, what did you mean?

- o = Object.new
- def o.each

```
# (2)
x = yield
p x           #=> "foo"
# (5)
x = yield
p x           #=> nil
# (7)
x = yield
# not reached
p x
```

- end
- e = o.to_enum

- **(1)**
- e.next

- **(3)**
- e.feed "foo"

- **(4)**
- e.next

- **(6)**

- e.next

- **(8)**

- three_times = Enumerator.new do |yielder|

- `3.times do |x|`
- `  result = yielder.yield(x)`
- `  puts result`
- `end`
- end
- three_times.next # => 0
- three_times.feed("foo")
- three_times.next # => 1, prints "foo"
- three_times.next # => 2, prints nothing

It became worse (to me).
(1) (2) (3)... meant the evaluation order. I believe that
this is helpful for newbie to understand it.

--
Yusuke Endoh mame@tsg.ne.jp

#3 - 06/01/2011 08:22 AM - drbrain (Eric Hodel)

Ryan and I worked on the example and wording to help clarify it since #feed is a difficult method to understand. Please double check our work.

#4 - 06/02/2011 12:23 AM - mame (Yusuke Endoh)

- ruby -v changed from ruby 1.9.3dev (2011-05-16 trunk 31583) [x86_64-darwin10.7.0] to -

The sample code looks very good! Thank you!
But it is hard for me to parse the explanation:

"Set the value for the next yield in the enumerator returns."

2011/6/1 Eric Hodel drbrain@segment7.net:

> Issue #4757 has been updated by Eric Hodel.
>
> ## Ryan and I worked on the example and wording to help clarify it since #feed is a difficult method to understand. Â Please double check our work.
>
> Bug #4757: Attempt to make Enumerator docs more clear (patch included)
> http://redmine.ruby-lang.org/issues/4757
>
> Author: David Copeland
> Status: Open
> Priority: Normal
> Assignee: Eric Hodel
> Category: DOC
> Target version:
> ruby -v: ruby 1.9.3dev (2011-05-16 trunk 31583) [x86_64-darwin10.7.0]
>
> The Enumerator docs aren't super clear; these make it more clear, at least to me, and also more consistent.
>
> --
> http://redmine.ruby-lang.org

--
Yusuke Endoh mame@tsg.ne.jp

#5 - 06/02/2011 06:23 AM - zenspider (Ryan Davis)

On Jun 1, 2011, at 08:20 , Yusuke ENDOH wrote:

> The sample code looks very good! Thank you!
> But it is hard for me to parse the explanation:
>
> "Set the value for the next yield in the enumerator returns."

totally agree... I just changed it to:

- Sets the value to be returned by the next yield inside +e+.
- If the value is not set, the yield returns nil.
- This value is cleared after being yielded.

#6 - 06/12/2011 08:23 PM - akr (Akira Tanaka)

2011/5/25 Yusuke Endoh mame@tsg.ne.jp:

> > - Note that enumeration sequence by next_values method does not affect other
> > - non-external enumeration methods, unless underlying iteration
> > - methods itself has side-effect, e.g. IO#each_line.
> > - Note that enumeration sequenced by +next_values+ does not affect other
> > - non-external enumeration methods, unless underlying iteration methods
> > - itself has side-effect, e.g. IO#each_line.
>
> I cannot undestand what "enumeration sequence by next_values" is.
> This is a problem of the original rdoc, not yours, though.
> It might be good to remove "enumeration sequence by".
>
> The old rdoc was written by akr. Akr, what did you mean?

For example, next_values doesn't affect Array#each as follows.

```
% ruby -e '
obj = [1,2,3,4]
enum = obj.each
obj.each {|x|
p x
enum.next_values
}
'
1
2
3
4
% ruby -e '
obj = [1,2,3,4]
enum = obj.each
obj.each {|x|
p x
}
'
1
2
3
4
```

The sequence (1, 2, 3, 4) is not changed by enum.next_values call.

But next_values affect IO#each as follows.

% print -l a b c d|ruby -e '

```
obj = STDIN
enum = obj.each
obj.each {|x|
p x
enum.next_values
}
'
"a\n"
"c\n"
% print -l a b c d|ruby -e '
obj = STDIN
enum = obj.each
obj.each {|x|
p x
}
'
"a\n"
"b\n"
"c\n"
"d\n"
```

Tanaka Akira

#7 - 06/12/2011 08:29 PM - mame (Yusuke Endoh)

Hello,

2011/6/12 Tanaka Akira akr@fsij.org:

> 2011/5/25 Yusuke Endoh mame@tsg.ne.jp:
>
> - Note that enumeration sequence by next_values method does not affect other
> - non-external enumeration methods, unless underlying iteration
> - methods itself has side-effect, e.g. IO#each_line.
> - Note that enumeration sequenced by +next_values+ does not affect other
> - non-external enumeration methods, unless underlying iteration methods
> - itself has side-effect, e.g. IO#each_line.
>
> I cannot undestand what "enumeration sequence by next_values" is.
> This is a problem of the original rdoc, not yours, though.
> It might be good to remove "enumeration sequence by".
>
> The old rdoc was written by akr. Â Akr, what did you mean?
>
> For example, next_values doesn't affect Array#each as follows.

Ah, I know the behavior. What I couldn't understand is just what
"enumeration sequence by next_values" is. Just English problem.

It is strange for me that the results returned by the method
("enumeration sequence by next_values") may affect Array#each.
I think that the subject that may affect Array#each is a method
("next_value"), in this case.

I can understand "next_values doesn't affect Array#each". So,
"enumeration sequence by" is not needed, right?

--

Yusuke Endoh mame@tsg.ne.jp

#8 - 06/12/2011 09:29 PM - akr (Akira Tanaka)

2011/6/12 Yusuke ENDOH mame@tsg.ne.jp:

> I can understand "next_values doesn't affect Array#each". So,
> "enumeration sequence by" is not needed, right?

I see.
It makes sense.

Tanaka Akira

#9 - 06/17/2011 08:00 AM - drbrain (Eric Hodel)

So should the final paragraph in #next_values be:

Note that +next_values+ does not affect other non-external enumeration methods unless the underlying iteration methods itself has a side effect, e.g. IO#each_line

#10 - 07/09/2011 09:25 AM - drbrain (Eric Hodel)

- Status changed from Open to Closed

This issue was solved with changeset r32466.
David, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- enumerator.c: Remove "enumeration sequenced by".
  [Ruby 1.9 - Bug #4757]

Files

0003-clean-up-can-clarify-Enumerator.patch	13.2 KB	05/22/2011	davetron5000 (David Copeland)