Ruby trunk - Feature #14044

Introduce a new attribute `step` in Range

10/23/2017 06:18 AM - mrkn (Kenta Murata)

Status:	Rejected
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	

Description

As described in #13904, Numo::NArray and PyCall touches internal structure of Enumerator to obtain the argument of Range#step.
This information is necessary to realize the same manner of the utilization of Python's slices.

In this ticket, I propose to introduce the new attribute step in an instance of Range class.
Its role is same as the step attribute of Python's slice.

Range#step should be changed to return a new Range object with the given step value instead of an Enumerator.
The default value of step attribute is nil, and it means that step is 1.

Related issues:

Related to Ruby trunk - Feature #13904: getter for original information of En...	**Closed**
Related to Ruby trunk - Feature #3714: Add getters for Enumerator	**Closed**
Related to Ruby trunk - Feature #9049: Shorthands (a:b, *) for inclusive inde...	**Open**

History

#1 - 10/23/2017 06:18 AM - mrkn (Kenta Murata)

- Related to Feature #13904: getter for original information of Enumerator added

#2 - 10/23/2017 11:11 AM - Eregon (Benoit Daloze)

That's a big breaking change, and Range then needs to know about something like a "step value" which makes it more complex.
Why not introducing your own StepRange or so in PyCall then?

BTW, I think #13904 is more reasonable,
and only makes information that is always needed anyway and already can be observed though Enumerator#inspect available for other uses.
e.g. if someone wants to pretty-print an Enumerator differently than #inspect, they should be able to get the receiver, method and arguments, no point to hide that, it's already user-visible in #inspect.

#3 - 10/23/2017 11:13 AM - Eregon (Benoit Daloze)

Could you show some example code where you want to use this?
I think that was missing from the other issue.

#4 - 11/07/2017 08:20 AM - mrkn (Kenta Murata)

I want this feature for specifying a sliced view of a multi-dimensional array.
In Python, it can be represented by using slice notation or a slice object:

```
>>> ary = numpy.arange(0, 36).reshape((6, 6))
>>> ary
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11],
       [12, 13, 14, 15, 16, 17],
       [18, 19, 20, 21, 22, 23],
       [24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35]])
>>> ary[1:5:2, slice(1, 5, 2)]
array([[ 7,  9],
       [19, 21]])
```

In the above example, 1:5:2 is a slice notation, and slice(1, 5, 2) is a slice object.
Both have the same mean: from 1 to 4 with step 2.

I want to write the same thing in Ruby, but the current Ruby doesn't have any notations for slice.

While I tried to use : symbol in Ruby as Python, but it failed because :: has the specific role.

Introducing step attribute in Range, we can get the value of step of a Range in a formal way, and we can write sliced views as following:

```
ary[(1..4).step(2), (1..4).step(2)]
```

#5 - 11/29/2017 03:06 AM - mrkn (Kenta Murata)

- Target version set to 2.6

- Assignee set to matz (Yukihiro Matsumoto)

#6 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)

#7 - 03/15/2018 12:24 AM - mrkn (Kenta Murata)

- Related to Feature #3714: Add getters for Enumerator added

#8 - 03/15/2018 12:24 AM - mrkn (Kenta Murata)

*- Related to Feature #9049: Shorthands (a:b, *) for inclusive indexing added*

#9 - 03/15/2018 06:31 AM - mrkn (Kenta Murata)

- Status changed from Open to Rejected

I've discussed this issue and #13904 in today's developer meeting.
Following the discussion, I decided to withdraw the proposal of range with step.
Instead I propose to introduce a subclass of Enumerator that provides the attributes of first, last, and step values of range with step.

I will close this issue and reopen #13904 for the new proposal.

Thanks.