Ruby trunk - Feature #15092

Provide step count in Range constructor

09/09/2018 02:28 AM - v0dro (Sameer Deshmukh)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	

Description

I would like to propose making changes to the Range constructor so that a user can specify a step count along with start and stop. Since Ruby 2.6 will introduce a step property in Ranges anyway I think this will be a useful addition.

Here's my reasons for the changes:

When creating software libraries for numerical computing, it is common to query the data container for values in a particular range at some given steps. For example, say I have the following NArray object:

```
a = NArray.new([1,2,3,4,5,6,7,8,9,10,11,12])
```

And I want the values 1, 4, 7, 10 and 12, I can simply specify a Range like this:

```
r = Range.new 0, Float::INFINITY, 3 # start, stop (upto the end), step
a[r]
# => NArray([1, 4, 7, 10, 12])
```

This can possibly also be extended to Array#[] so that users can get ranges of values at steps without much worry.

Related issues:

Is duplicate of Ruby trunk - Feature #13904: getter for original information ...	**Closed**

History

#1 - 09/09/2018 02:42 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Feedback

No need to change the constructor. Instead of:

```
Range.new 0, Float::INFINITY, 3
```

Use the shorter

```
(0...) % 3
```

#2 - 09/09/2018 02:59 AM - v0dro (Sameer Deshmukh)

Advantages of changing the constructor:

- Makes it easy to read test code for someone new to Ruby.
- Consistency in specifying step count in constructor (simple and straightforward way) and using the shorter syntax (idiomatic Ruby way).

Disadvantages:

- Few extra lines of code to change the constructor.

I think the advantages outweigh the disadvantages.

#3 - 09/09/2018 03:12 AM - duerst (Martin Dürst)

v0dro (Sameer Deshmukh) wrote:

> And I want the values 1, 4, 7, 10 and 12, I can simply specify a Range like this:
>
> ```
> r = Range.new 0, Float::INFINITY, 3 # start, stop (upto the end), step
> ```

```
a[r]
# => NArray([1, 4, 7, 10, 12])
```

Wouldn't the result be [1, 4, 7, 10, 13]?

#4 - 09/09/2018 03:36 AM - v0dro (Sameer Deshmukh)

> Wouldn't the result be [1, 4, 7, 10, 13]?

Ah yes. My bad. Editing the description. Thank you.

#5 - 09/09/2018 03:37 AM - v0dro (Sameer Deshmukh)

v0dro (Sameer Deshmukh) wrote:

> I would like to propose making changes to the Range constructor so that a user can specify a step count along with start and stop. Since Ruby 2.6 will introduce a step property in Ranges anyway I think this will be a useful addition.
>
> Here's my reasons for the changes:
>
> When creating software libraries for numerical computing, it is common to query the data container for values in a particular range at some given steps. For example, say I have the following NArray object:
>
> ```
> a = NArray.new([1,2,3,4,5,6,7,8,9,10,11,12])
> ```
>
> And I want the values 1, 4, 7, 10 and 12, I can simply specify a Range like this:
>
> ```
> r = Range.new 0, Float::INFINITY, 3 # start, stop (upto the end), step
> a[r]
> # => NArray([1, 4, 7, 10, 12])
> ```
>
> This can possibly also be extended to Array#[] so that users can get ranges of values at steps without much worry.

The array will return [1,4,7,10]. Sorry for previous mistake.

#6 - 09/09/2018 09:19 AM - shevegen (Robert A. Heiler)

> Wouldn't the result be [1, 4, 7, 10, 13]?

Off-by-one is ... common. :)

A bit more on topic, Float::INFINITY is quite long. Could we not use :infinity to refer to it in some methods or something shorter? I think ruby users should not need to have to know the leading "namespace" (Float) in order to refer to a concept of infinity in ruby.

#7 - 09/09/2018 01:57 PM - marcandre (Marc-Andre Lafortune)

v0dro (Sameer Deshmukh) wrote:

> Advantages of changing the constructor:
>
> - Makes it easy to read test code for someone new to Ruby.

Very debatable.

- Consistency in specifying step count in constructor (simple and straightforward way) and using the shorter syntax (idiomatic Ruby way).

It's actually not consistent, since the result is a Enumerator::ArithmeticSequence, not a Range.

> Disadvantages:

- Few extra lines of code to change the constructor.

I realize you wrote "introduces a step property in Ranges anyway I think this will be a useful addition", but this is not true. Range will not have a step property. It's simply that the step method will now return a different class of object that is more useful.

#8 - 09/11/2018 01:01 AM - mrkn (Kenta Murata)

- Is duplicate of Feature #13904: getter for original information of Enumerator added

#9 - 09/11/2018 01:03 AM - mrkn (Kenta Murata)

We've already have Emuerator::ArithmeticSequence in trunk.
Please use it.

#10 - 09/11/2018 01:16 AM - v0dro (Sameer Deshmukh)

OK I'm convinced this is probably a bad idea since step returns an ArithmeticSequence and is not a property of Range.

Closing the issue.

#11 - 09/11/2018 01:18 AM - mrkn (Kenta Murata)

- Status changed from Feedback to Rejected