# Ruby trunk - Feature #8506

## Object#iter_for / Object#to_iter

06/09/2013 10:31 PM - alindeman (Andy Lindeman)

| | | |
|---|---|---|
| **Status:** | Open | |
| **Priority:** | Normal | |
| **Assignee:** | | |
| **Target version:** | | |

### Description

=begin
Ruby's (({Enumerator})) and (({#enum_for})) methods are very powerful and I use them very often. However, (({Object#enum_for})) requires a method that yields, usually in some sort of loop.

Many objects in Ruby have methods that iterate to a "next value," but do not yield. For example, (({Fixnum#next.})) There is no way to use (({Fixnum#next})) with (({#enum_for})) directly that I am aware of.

I propose the introduction of (({Object#iter_for})) which--given a method--generates a lazy sequence by continually invoking the method on successive values. I call it (({iter})) or (({iterate})) because it is very similar to clojure's iterate:
http://clojure.github.io/clojure/clojure.core-api.html#clojure.core/iterate

Proposed API:

0.iter_for(:next).take(5) # => [0, 1, 2, 3, 4]

require 'date'
Date.new(2013, 1, 1).iter_for(:next_month).take(3) # => [Tue, 01 Jan 2013, Fri, 01 Feb 2013, Fri, 01 Mar 2013]

I am especially excited about (({0.iter_for(:next)})) as I find myself using infinite lazy numeric sequences more often lately to solve specific kinds of problems. Right now you are required to write something like: (({Enumerator.new { |y| i = 0; loop { y << i; i += 1 } }})) or (({(0..Float::INFINITY).each})). Neither is especially elegant or happy to the developers' eyes in my opinion.

Thank you all :) Ruby is an amazing tool.
=end

### History

#### #1 - 06/09/2013 10:34 PM - charliesome (Charlie Somerville)

+1

#### #2 - 06/10/2013 12:38 AM - Anonymous

Btw., regarding Qbject#to_enum, what is your opinion? Do you use it often? Or is there something about it that makes it less useful?

#### #3 - 06/10/2013 02:44 PM - nobu (Nobuyoshi Nakada)

*- Description updated*

You may want to show the implementation in ruby (and tests)?

#### #4 - 06/10/2013 05:20 PM - Eregon (Benoit Daloze)

Here is the related blogpost: http://alindeman.github.io/2013/06/10/porting-iterate-to-ruby.html

#### #5 - 06/10/2013 05:21 PM - phluid61 (Matthew Kerwin)

nobu (Nobuyoshi Nakada) wrote:

> You may want to show the implementation in ruby (and tests)?

Here is an implementation: https://gist.github.com/phluid61/5747216