Ruby - Feature #1119

with_index_from

02/06/2009 06:40 PM - matz (Yukihiro Matsumoto)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

=begin
まつもと ゆきひろです

In message "Re: [ruby-dev:37928] Re: [Feature:trunk] with_index_from"
on Fri, 6 Feb 2009 12:04:30 +0900, "Akinori MUSHA" knu@iDaemons.org writes:

|□しかし、 each_with_index_from という名前は長すぎると思います。
|また each_with_index 自体、 Enumerator が導入された今となっては
|each.with_index よりも若干実行効率がよいことを除けば存在意義が
|薄いですね。
|
|□ここは、 with_index にだけ引数を追加して each_with_index は
|そのままにしてはどうでしょう。
|
|StringIO.new("foo|bar|baz").each("|").with_index(1) { ... }
|
|□ちょっと凝ったことをしたければ Enumerator 経由で行ってね、
|というスタンスは悪くはないと思います。

each_with_indexは放置。Enumerator#with_indexの方で手当てする
方に賛成します。が、現時点では簡潔なwith_index(1)にするのか、
より明確なwith_index_from(1)にするのかは決断できません。どっ
ちがよいでしょうねえ。
=end

History

#1 - 02/07/2009 12:50 AM - mame (Yusuke Endoh)

=begin
遠藤です。

2009/2/6 Yukihiro Matsumoto matz@ruby-lang.org:

> まつもと ゆきひろです
>
> In message "Re: [ruby-dev:37928] Re: [Feature:trunk] with_index_from"
> on Fri, 6 Feb 2009 12:04:30 +0900, "Akinori MUSHA" knu@iDaemons.org writes:
>
> |□しかし、 each_with_index_from という名前は長すぎると思います。
> |また each_with_index 自体、 Enumerator が導入された今となっては
> |each.with_index よりも若干実行効率がよいことを除けば存在意義が
> |薄いですね。
> |
> |□ここは、 with_index にだけ引数を追加して each_with_index は
> |そのままにしてはどうでしょう。
> |
> |StringIO.new("foo|bar|baz").each("|").with_index(1) { ... }
> |
> |□ちょっと凝ったことをしたければ Enumerator 経由で行ってね、
> |というスタンスは悪くはないと思います。

なるほど！ 感服しました。

> each_with_indexは放置。Enumerator#with_indexの方で手当てする
> 方に賛成します。が、現時点では簡潔なwith_index(1)にするのか、

より明確なwith_index_from(1)にするのかは決断できません。どっちがよいでしょうねえ。

with_index_from はやむなく考えたものなので、with_index にだけ引数を持たせていいならそっちに一票投じます。

with_index だけ引数を持たせるパッチです。

Index: enumerator.c

--- enumerator.c (revision 22101)
+++ enumerator.c (working copy)
@@ -406,32 +406,51 @@

/*

- call-seq:
  - e.with_index {|(*args), idx| ... }
  - e.with_index(offset = 0) {|(*args), idx| ... }
  - e.with_index
  - Iterates the given block for each element with an index, which
  - start from 0. If no block is given, returns an enumerator.
  - start from +offset+. If no block is given, returns an enumerator.

*/
static VALUE
-enumerator_with_index(VALUE obj)
+enumerator_with_index(int argc, VALUE *argv, VALUE obj)
{
struct enumerator *e;

- VALUE memo = 0;
- int argc = 0;
- VALUE *argv = 0;

- VALUE memo;

- RETURN_ENUMERATOR(obj, 0, 0);

- rb_scan_args(argc, argv, "01", &memo);
- RETURN_ENUMERATOR(obj, argc, argv);
- memo = NIL_P(memo) ? 0 : (VALUE)NUM2LONG(memo);
e = enumerator_ptr(obj);
if (e->args) {
argc = RARRAY_LEN(e->args);
argv = RARRAY_PTR(e->args);
}
- else {
- argc = 0;
- argv = NULL;
- }
return rb_block_call(e->obj, e->meth, argc, argv,
enumerator_with_index_i, (VALUE)&memo);
}

+/*

  - call-seq:
  - e.each_with_index {|(*args), idx| ... }
  - e.each_with_index
  - Same as Enumeartor#with_index, except each_with_index does not
  - receive an offset argument.

- */
  static VALUE
  +enumerator_each_with_index(VALUE obj)
  +{
- return enumerator_with_index(0, NULL, obj);
  +}

+static VALUE
enumerator_with_object_i(VALUE val, VALUE memo)
{
return rb_yield_values(2, val, memo);
@@ -841,9 +860,9 @@
rb_define_method(rb_cEnumerator, "initialize", enumerator_initialize, -1);
rb_define_method(rb_cEnumerator, "initialize_copy",
enumerator_init_copy, 1);
rb_define_method(rb_cEnumerator, "each", enumerator_each, 0);

- rb_define_method(rb_cEnumerator, "each_with_index",
  enumerator_with_index, 0);

- rb_define_method(rb_cEnumerator, "each_with_index",
  enumerator_each_with_index, 0);
  rb_define_method(rb_cEnumerator, "each_with_object",
  enumerator_with_object, 1);

- rb_define_method(rb_cEnumerator, "with_index", enumerator_with_index, 0);

- rb_define_method(rb_cEnumerator, "with_index", enumerator_with_index, -1);
  rb_define_method(rb_cEnumerator, "with_object", enumerator_with_object, 1);
  rb_define_method(rb_cEnumerator, "next", enumerator_next, 0);
  rb_define_method(rb_cEnumerator, "rewind", enumerator_rewind, 0);

--
Yusuke ENDOH mame@tsg.ne.jp

=end

#2 - 02/12/2009 05:00 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

=begin
separated from #1112 unintentionally.
=end