Ruby - Bug #12466

Enumerable should yield multiple values when possible

06/07/2016 09:23 AM - sos4nt (Stefan Schüßler)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.3.1	**Backport:**	2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

Some methods in Enumerable and Array yield arrays instead of multiple values, e.g.:

```
[1, 2, 3, 4].each_cons(2).peek_values #=> [[1, 2]]
```

If each_cons would yield multiple values (i.e. yield 1, 2 instead of yield [1, 2]), we could write:

```
[1, 2, 3, 4].each_cons(2).map(&:quo) #=> [(1/2), (1/3), (1/4)]
```

But currently, this results in an exception and we have to provide a block to achieve the above:

```
[1, 2, 3, 4].each_cons(2).map { |a, b| a.quo(b) } #=> [(1/2), (2/3), (3/4)]
```

No big deal, but it looks a bit cumbersome (in Ruby terms) and I don't see how the current behavior is preferable.

Another example is reverse_each which cripples yielded values:

```
def each_two_values
  return enum_for(__method__) unless block_given?
  yield 1, 2
  yield 2, 3
  yield 3, 4
end

each_cons_values.map(&:quo) #=> [(1/2), (2/3), (3/4)]

each_cons_values.reverse_each.map(&:quo)
#=> NoMethodError: undefined method `quo' for [3, 4]:Array
```

Of course, I can easily provide my own implementation:

```
module Enumerable
  def my_reverse_each
    return enum_for(__method__) unless block_given?
    map { |*values| values }.reverse.each { |values| yield *values }
  end
end

each_cons_values.my_reverse_each.map(&:quo) #=> [(3/4), (2/3), (1/2)]
```

But shouldn't this be the default behavior?

If an array is actually needed (instead of multiple values), there's Enumerable#each_entry which performs the conversion.

BTW, other methods *do* yield multiple values:

```
%w(a b c).each.with_index(1).peek_values #=> ["a", 1]
```

History

#1 - 07/19/2016 09:04 AM - nobu (Nobuyoshi Nakada)

- Description updated

IIRC, these methods yielded multiple values once, but reverted due to the backward compatibilities.

#2 - 08/09/2016 06:42 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

This idea sounds nice, but we have to keep compatibility.

Matz.