

Ruby - Feature #10880

Subclassing Array from Enumerator

02/21/2015 08:04 PM - sawa (Tsuyoshi Sawada)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description		
It often happens that I want to use a method from the Enumerator class on an array, and so I frequently use Array#to_enum. Since Array and Enumerator share the same parent Enumerable, they should be very close. I think it would be convenient if Array inherits from Enumerator. (Perhaps the same thing can be said for Hash too)		

History

#1 - 02/22/2015 02:24 AM - marcandre (Marc-Andre Lafortune)

they should be very close

They *are* very close:

```
Enumerator.instance_methods - Array.instance_methods
# => [:with_index, :with_object, :next_values, :peek_values, :next, :peek, :feed, :rewind]
```

Which method(s) in there do you claim to use frequently?

#2 - 02/22/2015 05:14 AM - sawa (Tsuyoshi Sawada)

Marc-Andre Lafortune wrote:

```
Enumerator.instance_methods - Array.instance_methods
# => [:with_index, :with_object, :next_values, :peek_values, :next, :peek, :feed, :rewind]
```

Which method(s) in there do you claim to use frequently?

I use `with_index` and `with_object` frequently.

#3 - 02/22/2015 03:09 PM - marcandre (Marc-Andre Lafortune)

I use `with_index` and `with_object` frequently.

Any reason why you wouldn't simply use `each_with_index` and `each_with_object`?

#4 - 02/22/2015 03:13 PM - sawa (Tsuyoshi Sawada)

Marc-Andre Lafortune wrote:

I use `with_index` and `with_object` frequently.

Any reason why you wouldn't simply use `each_with_index` and `each_with_object`?

I want to provide an optional starting number (most often 1) to the former.

But you are making me realize that my motivation for the proposal is getting weaker.

#5 - 01/04/2020 06:54 PM - sawa (Tsuyoshi Sawada)

I should use `each_with_index` and `each_with_object` as marcandre suggests. I withdraw this. Please close this issue.

#6 - 01/04/2020 08:24 PM - Eregon (Benoit Daloze)

- Status changed from Open to Rejected