

Ruby master - Feature #10482

Allow ignored items to vary in `Enumerable#chunk`.

11/06/2014 07:56 AM - sawa (Tsuyoshi Sawada)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
In #5663 , regarding a method proposed, Yehuda Katz's writes:	
The only caveat is that it would be impossible to intentionally return nil here; suggestions welcome.	
I would like to note here that the same problem exists with Enumerable#chunk. Currently, when the key value is nil, the corresponding items are thrown out. That may be useful sometimes, but sometimes, silently doing so causes a hard-to-detect bug. At least, there should be a way to change what is ignored (which would not break existing code using it), and ideally, nothing should be thrown out unless explicitly specified (which would break existing code).	
I propose Enumerable#chunk to take an optional named parameter ignore, which switches what is ignored. When something other than nil is specified, then nil should not be ignored:	
<pre>[:foo1, :foo2, "bar", nil, nil].chunk(ignore: String){ e e.class} # => [[Symbol, [:foo1, :foo2]], [NilClass, [nil, nil]]]</pre>	
When you don't want anything to be ignored, then the parameter should be set to something that does not appear in the receiver:	
<pre>[:foo1, :foo2, "bar", nil, nil].chunk(ignore: "nothing to ignore"){ e e.class} # => [[Symbol, [:foo1, :foo2]], [String, ["bar"]], [NilClass, [nil, nil]]]</pre>	

History

#1 - 11/06/2014 08:01 AM - sawa (Tsuyoshi Sawada)

Sorry, the example was wrong; I mistook nil and NilClass. But I hope you get the point.

#2 - 11/11/2014 10:08 AM - akr (Akira Tanaka)

- Status changed from Open to Feedback

The method has a form which is incompatible with your proposal: `enum.chunk(initial_state) { |elt, state| ... }`

However I deprecated the form (since Ruby 2.2), your proposal may be considerable in future.