

Ruby master - Feature #12813

Calling `chunk_while`, `slice_after`, `slice_before`, `slice_when` with no block

10/06/2016 04:43 AM - marcandre (Marc-Andre Lafortune)

Status:	Assigned	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
Currently, <code>chunk_while</code> , <code>slice_after</code> , <code>slice_before</code> , <code>slice_when</code> all require a block.		
If one needs the index within the block, there is no good way to do this; <code>enum.each_with_index.chunk_while</code> would have indices in the results, so <code>enum.enum_for(:chunk_while).with_index</code> is the best solution.		
I feel that we should return <code>enum_for(:chunk_while)</code> . This is strictly more useful than raising as we currently do.		

History

#1 - 11/25/2016 02:04 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

We briefly looked at this issue in today's developer meeting but had no time to fully agree that the proposed extension is safe. We just did not reach a conclusion.

#2 - 11/25/2016 09:40 PM - marcandre (Marc-Andre Lafortune)

It would help if you explained what is meant by "safe".

There is currently no valid usage of `chunk_while`, `slice_after`, `slice_before` or `slice_when` without a block and an argument, as they all raise an exception. So "extending" them by returning an enumerator instead can't reasonably produce a problem. So what else could be meant by "safe"?

#3 - 12/06/2016 12:58 PM - matz (Yukihiro Matsumoto)

After consideration, I accept this proposal.

Matz.