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)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

Description
Currently, chunk_while, slice_after, slice_before, slice_when all require a block.

If one needs the index within the block, there is no good way to do this; enum.each_with_index.chunk_while would have indices in the results, so enum.enum_for(:chunk_while).with_index is the best solution.

I feel that we should return enum_for(:chunk_while). 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 todays 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.