

Ruby master - Feature #12165

Hash#first, Hash#last

03/10/2016 06:00 PM - stillhart (Fabian Stillhart)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
Description	
Just run into a simple problem with a colleague and was wondering why there is no Hash#last method?	
<pre>{a: true, b: false}.last NoMethodError: undefined method 'last' for {:a=>true, :b=>false}:Hash</pre>	
Interestingly I while playing around I found out that the Hash#first method works. But why is it not in the ruby-doc?	
<pre>{a: true, b: false}.first => [:a, true]</pre>	
I would assume the Hash#last method would work like the Hash#first method:	
<pre>{a: true, b: false}.last => [:b, false]</pre>	
If I am not wrong the order of a Hash is always the same when calling Hash#each. So wouldn't it make sense to have Hash#last method?	
I tested it in Ruby 2.2.4 and Ruby 2.3.0.	

History

#1 - 03/12/2016 10:55 AM - shevegen (Robert A. Heiler)

Yes, I would concur considering that ruby hashes are ordered these days.

It may have been different in the past when the hashes were not ordered.

But the moment when ruby hashes were ordered, I think using .first or .last would make sense now. (I do remember that apeiros on IRC once said that one should still not rely on Hashes being ordered; no idea why or why not, just thought I would mention it.)

The method .last() also seems to work for Strings too:

```
`"abc".last # => "c`
```

The way I do this usually for Hashes is by first calling .keys, then .first or .last respectively, but a direct method may be more convenient. (Not that it would be a big deal for me either way, I can live without it too.)

A possible reason may be that it may be not specific, e. g. if you want the first key or last key, or first key and last value, but this could be solved too, such as .first_value or .last_value if you know what I mean?

But I assume that, most everyone wants to just get the .first key and .last key anyway since it is the main pointer.

However had, as your example shows, you want .last and .first to return both key and value for a Hash. Which is also fine for me, I don't mind either way, just that it has to be documented properly. :)