

Ruby master - Feature #15747

`(..1).last(2)` should return array but raise TypeError

04/04/2019 12:12 AM - wanabe (_ wanabe)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	
Description	
Range#last with argument for beginless range shouldn't raise exception, should it?	
<pre>\$./miniruby -ve 'p (..1).last(2)' ruby 2.7.0dev (2019-04-04 trunk 67430) [x86_64-linux] Traceback (most recent call last): 4: from -e:1:in `<main>' 3: from -e:1:in `last' 2: from -e:1:in `to_a' 1: from -e:1:in `to_a' -e:1:in `each': can't iterate from NilClass (TypeError)</main></pre>	
Range#last without argument works fine.	
<pre>\$./miniruby -ve 'p (..1).last' ruby 2.7.0dev (2019-04-04 trunk 67430) [x86_64-linux] 1</pre>	

History

#1 - 04/04/2019 12:48 AM - mame (Yusuke Endoh)

- Status changed from Open to Feedback

Thank you for playing with a beginless range.

In general, I think that an enumerating operation against a beginless range should raise an exception. It is also the case at Range#last with an argument. Note that the meaning of Range#last with no argument is fairly different from the version with a length argument. See [#8739](#) and [#15523](#).

That said, Range#last might be handled as a special case. I have no strong opinion about this, but an exception is still one of the most reasonable design choices.

1) It is difficult to reversely enumerate the elements of Range from last. AFAIK, there is no precedent.

2) (-Float::INFINITY .. 1).last(2) also raises a TypeError.

```
(-Float::INFINITY .. 1).last(2) #=> can't iterate from Float (TypeError)
```

#2 - 04/04/2019 01:02 AM - wanabe (_ wanabe)

- Backport deleted (2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN)

- ruby -v deleted (ruby 2.7.0dev (2019-04-04 trunk 67430) [x86_64-linux])

- Tracker changed from Bug to Feature

Thank you to response.

mame (Yusuke Endoh) wrote:

In general, I think that an enumerating operation against a beginless range should raise an exception. It is also the case at Range#last with an argument. Note that the meaning of Range#last with no argument is fairly different from the version with a length argument. See [#8739](#) and [#15523](#).

Oh, I've missed [#8739](#) and [#15523](#). Thank you.

That said, Range#last might be handled as a special case. I have no strong opinion about this, but an exception is still one of the most reasonable design choices.

1) It is difficult to reversely enumerate the elements of Range from last. AFAIK, there is no precedent.

2) `(-Float::INFINITY .. 1).last(2)` also raises a `TypeError`.

```
(-Float::INFINITY .. 1).last(2) #=> can't iterate from Float (TypeError)
```

OK. I have no strong opinion, too.

I have realized this is a not "Bug". I change the tracker "Feature".

#3 - 10/06/2019 06:30 AM - wanabe (_ wanabe)

- Status changed from Feedback to Rejected

Now, I think this proposal doesn't make sense.

So I reject it. thank you.