Ruby master - Feature #15813
Proposal: Add exception support in `Range#first`
04/30/2019 02:52 PM - osyo (manga osyo)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version</td>
<td></td>
</tr>
</tbody>
</table>

### Description

**Current status**

Calling `Range#last` in endless range((1..)) raises an exception.

```ruby
# OK
p (1..Float::INFINITY).end     # => Infinity
p (1..).end                   # => nil
p (1..Float::INFINITY).last   # => Infinity

# NG: Raise error: in `last`: cannot get the last element of endless range (RangeError)
p (1..).last
p (1..).last(1)
```

But, calling `Range#first` in beginless range((..1)) does not raise an exception.

```ruby
# OK
p (-Float::INFINITY..1).begin  # => -Infinity
p (..1).begin                  # => nil
p (-Float::INFINITY..1).first  # => -Infinity

# OK: Does not raise
p (..1).first                  # => nil

# NG: Raise error: in `each`: can't iterate from NilClass (TypeError)
p (..1).first(1)
```

I think the current situation is not consistent, so it is necessary to move the behavior to one side or the other. Also, in the case of `Range#last`, an exception is explicitly raised.

See: [https://github.com/ruby/ruby/blob/6a3165e19dfa21babfb2ef1f1c20c9930410b0ec/range.c#L1100-L1102](https://github.com/ruby/ruby/blob/6a3165e19dfa21babfb2ef1f1c20c9930410b0ec/range.c#L1100-L1102)

### Proposal

Added support to raise an exception for `Range#first` too.

**Before**

```ruby
# OK
p (-Float::INFINITY..10).begin  # => -Infinity
p (..10).begin                 # => nil
p (-Float::INFINITY..10).first  # => -Infinity
p (..10).first                 # => nil
p Range.new(nil, 10).first     # => nil

# NG: Raise error: in `each`: can't iterate from NilClass (TypeError)
p (..10).first(1)
```

**After**

```ruby
# OK
p (-Float::INFINITY..10).begin  # => -Infinity
p (..10).begin                 # => nil
p (-Float::INFINITY..10).first  # => -Infinity
```
# NG: Raise error: in `first': cannot get the first element of beginless range (RangeError)
```
p (..10).first
p Range.new(nil, 10).first
p (..10).first(1)
```
# in Ruby 2.6.1
```
p Range.new(nil, 10).first
# Error: in `initialize': bad value for range (ArgumentError)
```

Thank you.

pull request: [https://github.com/ruby/ruby/pull/2163](https://github.com/ruby/ruby/pull/2163)

---

**History**

#1 - 04/30/2019 03:06 PM - mame (Yusuke Endoh)
Merged at 4e88e8692844a2a317bc19481f0f2601b6f00955. Thank you.

#2 - 04/30/2019 03:06 PM - mame (Yusuke Endoh)
- Status changed from Open to Closed

#3 - 04/30/2019 03:10 PM - osyo (manga osyo)
Thanks!!!!!!