

Ruby master - Bug #12090

`Range#cover` does not raise an exception when comparison fails

02/19/2016 03:37 PM - sawa (Tsuyoshi Sawada)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

The documentation for `Range#cover?` says,

```
cover?(obj) → true or false
```

Returns true if obj is between the begin and end of the range.

```
This tests begin <= obj <= end when exclude_end? is false and begin <= obj < end when exclude_end? is true.
```

which implies that, when comparison fails, `cover?` should raise an error. But actually, it doesn't:

```
"a" <= 1 # => ArgumentError: comparison of String with 1 failed
1 <= "a" # => ArgumentError: comparison of Fixnum with String failed
("a.."z").cover?(1) # => false
```

It silently returns false instead of raising an error.

Either `Range#cover?` should raise an error when comparison fails, or the documentation should be changed to correctly describe this behaviour.

Associated revisions

Revision 3556cba5 - 07/24/2019 05:35 PM - jeremyevans (Jeremy Evans)

Document that `Range#cover?` returns false if `<=>` returns nil

Fixes [Bug #12090]

History

#1 - 02/19/2016 05:01 PM - sos4nt (Stefan Schüßler)

Raising an error would break backward compatibility. Maybe `cover?` should just return nil to indicate incomparable values.

```
("a.."z").cover?("c") # => true
("a.."z").cover?("5") # => false
("a.."z").cover?(1) # => nil
```

#2 - 07/24/2019 05:36 PM - jeremyevans (Jeremy Evans)

- Status changed from Open to Closed

Applied in changeset [git|3556cba5038530e728eb6309dcf5d4d1a96a02ac](https://github.com/ruby/ruby/commit/3556cba5038530e728eb6309dcf5d4d1a96a02ac).

Document that `Range#cover?` returns false if `<=>` returns nil

Fixes [Bug #12090]