Please comment your favorite ticket numbers you want to ask to discuss with your SHORT comment or summary. (your summary/comment will help us because we don't need to read all of ticket comments)

**DO NOT** discuss then on this ticket, please.

Date: 2019/01/10 (Thu)
Time: 13:00-17:00 (JST)
Place: Cookpad Inc. (Tokyo, Japan)
Sign-up: https://ruby.connpass.com/event/113278
log: https://docs.google.com/document/u/1/d/1W_wrFsFxxU1MepA6HBpfl-9h-nLD8EBsgMsS7yTDvHE/pub
logedit: https://docs.google.com/document/u/1/d/1W_wrFsFxxU1MepA6HBpfl-9h-nLD8EBsgMsS7yTDvHE/edit

**NOTES**
- Dev meeting *IS NOT* a decision making place. All decisions should be done at the bug tracker.
- Dev meeting is a place we can ask Matz, nobu, nurse and other developers directly.
- Matz is a very busy person. Take this opportunity to ask him. If you can not attend, other attendees can ask instead of you (if attendees can understand your issue).
- We will write a log about discussion to a file or to each ticket in English.
- All activities are best-effort (keep in mind that most of us are volunteer developers).
- The date, time and place is scheduled according to when/where we can reserve Matz's time.

**Agenda**

Next dev-meeting

About 2.7 timeframe

Check security tickets

Carry-over from previous meeting(s)

From Attendees

From non-attendees

Please comment your favorite ticket we need to discuss with the following format.

* [Ticket ref] Ticket title (your name)
  * your comment why you want to put this ticket here if you want to add.

Your comment is very important if you are no attendee because we can not ask why you want to discuss about it.

Example:

* [Feature #14609] `Kernel#p` without args shows the receiver (ko1)
  * I feel this feature is very useful and some people say :+1: so let discuss about this feature.

I don't guarantee to put tickets in agenda if the comment violate the format (because it is hard to copy&paste).
**Related issues:**
Related to Ruby master - Misc #14770: [META] DevelopersMeeting

**Open**

---

**History**

#1 - 12/25/2018 12:26 PM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 12/25/2018 12:29 PM - mame (Yusuke Endoh)
- Description updated

#3 - 12/25/2018 12:30 PM - mame (Yusuke Endoh)
- [Bug #15404] Endless range has inconsistent chaining behaviour
  - Can we prohibit (1...1) and (1..1)..1 as a SyntaxError?

#4 - 12/25/2018 05:21 PM - aycabta (aycabta.)
- [Feature #14799] Startless range

#5 - 12/26/2018 01:46 AM - shyouhei (Shyouhei Urabe)
- [Bug #15460] Behaviour of String#setbyte changed
  - What is the expected / desired way to fix this?

#6 - 12/26/2018 08:08 AM - mame (Yusuke Endoh)
- [Bug #7300] Hash[] 1.9.3  ללשה
  - I think we can remove the compatibility layer for 1.9.0. Hash[[nil]] #=> {}

#7 - 12/27/2018 04:09 PM - k0kubun (Takashi Kokubun)
- [Misc #15347] Require C99
  - Is it okay to require C99 features which work on both VC12 (Visual Studio 2013) and Oracle Developer Studio 12 from Ruby 2.7?
  - Note: 2 major concerns (mswin, solars) seem to be solved
    - mswin platform maintainer usa (Usaku NAKAMURA) agreed to require VC12 from Ruby 2.7, which is tested on RubyCI and AppVeyor.
    - On Solaris, we support Oracle Developer Studio 12+ and GCC https://bugs.ruby-lang.org/issues/14200#note-5. We're already testing GCC on RubyCI, and planning to include Oracle Developer Studio 12.4 / 12.5 on RubyCI Solaris.

#8 - 12/28/2018 06:06 AM - mame (Yusuke Endoh)
- [Feature #6012] Proc#source_location also return the column
  - Now we can implement the issue.

#9 - 12/29/2018 10:41 AM - aycabta (aycabta.)
- [Feature #15483] Proc or Method combination with Symbol

#10 - 12/29/2018 12:52 PM - zverok (Victor Shepelev)

Old tickets with no reaction/vague reaction:
- [Feature #14444] MatchData: alias for #[]
- [Feature #14784] Comparable#clamp with a range
- [Feature #14145] Better Method#inspect

New tickets:
- [Bug #15428] Refactor Proc#>> and #<< (it regards the same problem as #15483 above, but approaches it from a different point of view)
- [Feature #15485] Refactor String#split
- [Misc #15486] Default gems README.md
- [Misc #15487] Clarify default gems maintenance policy

(Sorry for a huge list)
#11 - 12/30/2018 02:17 AM - osyo (manga osyo)

- [Feature #15373] Proposal: Enable refinements to #method and #instance_method
- [Feature #15374] Proposal: Enable refinements to #method_missing
- [Bug #15418] 配列リテラル内の引数を伴う括弧なしのメソッド呼び出しで syntax error

#12 - 01/02/2019 12:27 PM - Eregon (Benoit Daloze)

- [Bug #15497] Encoding of error messages should not depend on the locale encoding (eregon)
  - The behavior is unintuitive and makes it hard to test exception messages.
- [Bug #2250] IO::for_fd() objects' finalization dangerously closes underlying fds (eregon)
  - The current behavior is dangerous and has caused many spurious test/spec failures which are hard to find and debug. I plan to submit a patch, but I'd like opinions regarding compatibility.

#13 - 01/02/2019 02:10 PM - osyo (manga osyo)

- [Bug #15488] const_defined?("File::NULL") 
  - What is the expected behavior?

#14 - 01/09/2019 01:52 AM - mrkn (Kenta Murata)

- [Feature #15445] Reject '.123' in Float() method
  - I want to know Matz's opinion.

#15 - 01/09/2019 01:30 PM - mrkn (Kenta Murata)

- [Bug #15500] Behavior of require method in 2.5 is different from 2.4 and 2.6
  - Is this bug or not?

#16 - 01/09/2019 07:13 PM - alanwu (Alan Wu)

- [Feature #15456] Adopt some kind of consistent versioning mechanism
  - More uniformity within the ecosystem would be nice.

#17 - 01/10/2019 04:31 AM - mame (Yusuke Endoh)

- [Feature #11473] Immutable String literal in Ruby 3
  - As I recall, matz said that this has been cancelled at the old developers' meeting. I'd like to confirm.

#18 - 01/10/2019 07:50 AM - mame (Yusuke Endoh)

- [Feature #15477] Proc#arity returns -1 for composed lambda Procs of known arguments
  - May I commit?

#19 - 01/10/2019 09:53 PM - palkan (Vladimir Dementyev)

- [Feature #14344] refine at class level
  - there are some proposals for shorter refinements syntax; want to hear committers opinions (which one is better and good to be implemented)

#20 - 01/11/2019 07:52 PM - duerst (Martin Dürst)

On 2019/01/11 06:53, dementiev.vm@gmail.com wrote:

Issue #15462 has been updated by palkan (Vladimir Dementyev).

- [Feature #14344] refine at class level
  - there are some proposals for shorter refinements syntax; want to hear committers opinions (which one is better and good to be implemented)

You posted this at Japan time January 11, 6:53am. Please note that the meeting is in Japan and therefore on Japan time, which is 9 hours ahead of UTC. So if you want to raise an issue for this meeting, please make sure to do it one day in advance in your time zone.

Regards, Martin.
Could someone add a date + issue for the next developer meeting, please?

I looked at https://bugs.ruby-lang.org/projects/ruby-trunk/issues for when the next meeting may be, but was unable to find it (assuming so it may be in February 2019 but of course I do not know matz' schedule).

Thanks.

---

- Status changed from Open to Closed
- File Screenshot from 2019-01-19 09-17-17.png added

Here you are https://bugs.ruby-lang.org/issues/15546

Off topic: can someone give me the grant to copy issues around? It seems disabled for this account.

---

Enabled it to Developer role now.

---

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screenshot from 2019-01-19 09-17-17.png</td>
<td>71.9 KB</td>
<td>01/19/2019</td>
<td>shyouhei (Shyouhei Urabe)</td>
</tr>
</tbody>
</table>