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: 2018/10/10 (Wed)
Time: 14:00-18:00 (JST)
Place: pixiv Inc. (Tokyo, Japan)
Sign-up: https://ruby.connpass.com/event/101832/
log: https://docs.google.com/document/d/18f8peqDQ_nkLapK3BWFaAkEHaAlvD6hP0lhdkgDqTJl/pub
logedit: https://docs.google.com/document/d/18f8peqDQ_nkLapK3BWFaAkEHaAlvD6hP0lhdkgDqTJl/edit
Past meetings: https://bugs.ruby-lang.org/projects/ruby/wiki#Developer-Meetings

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 2.6 timeframe

Carry-over from previous meeting(s)

From Attendees

- [Feature #14839] How to deal with capitalizing Georgian in Unicode 11.0.0 (duerst)
  - I need feedback on this to be able to implement in in time for the Ruby 2.6 release.
- [Feature #15195] How to deal with new Japanese era (duerst)
  - We should prepare early (even if it’s just to check that we need to do nothing)
- How to address increasing spam to the bug tracker. (duerst)
  - #15212/#15213 are just two examples. They get removed (return a 404), which is good. But they reach the mailing list and its subscribers, which is a problem. Prefiltering bugs with URIs in titles seems to be a good start.
- [Misc #14632] [ANN] git.ruby-lang.org (hsbt)

From non-attendees

- [Feature #15123] Enumerable#compact proposal (greggzst)
  - It simplifies working with large and small collections so one doesn't have to remember that can't use #compact when enumerator is returned and have to fall back to #reject(nil?).
- [Feature #15112] Introduce the new singleton method STDERR.p (shevegen)
  - I am mostly curious what the ruby core team thinks about Kenta Murata's proposal; it probably will not take too much time away discussing it briefly, since the scope is small.
Comment format

- [Feature #11505] Module#=== should call #kind_of? on the object rather than rb_obj_is_kind_of which only searches the ancestor heirarchy (rafaelfranca)
  - This would allow patterns as Decorator and Proxy to work with case statements.
- [Feature #14912] Introduce pattern matching syntax (greggzst)
  - Many modern languages have introduced pattern matching. I used it in scala and found it very easy to utilize and understand especially in recursion. It makes extracting data easier as well.
- [Feature #15144] Enumerator#chain (zverok)

@zverok (Victor Shepelev) wants to hear about stale issues:

I am not sure if it is appropriate, but I'd also be very glad to hear about some "stale" discussions. They were typically reacted on developer meetings as "in general, good proposal (but not sure when it would be implemented/not sure about the name)", or something like that, and I'd like to know maybe we should do something to push them further? List of tickets:

- [Feature #14799] Startless range: usefullness discussed, patch provided by mame (Yusuke Endoh), waits for Matz's decision (?)
- [Feature #14784] Comparable#clamp accepting range: comment for akr (Akira Tanaka) about "needlessly big" proposal, I answered it, is it makes the proposal more likely to be accepted?
- [Feature #6284] Composition for procs: the last thing Matz has said is the operators are chosen (<< and >>), and "We need more discussion if we would add combination methods to the Symbol class." Is there a chance proc composition would make it way in the 2.6?
- [Feature #13581] Syntax sugar for method reference. The last thing Matz have said is: ":." looks best to me (followed by :::). Let me consider it for a while." Are there any choices made? Could we expect this for 2.6?
- [Feature #14781] Enumerator#generate It seems like people feel cautious enthusiasm about it, but not sure about the name. What should be done here? Voting on the name? Providing the patch with some name, and then voting for the name?

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 - 09/16/2018 01:22 PM - mame (Yusuke Endoh)
  - Related to Misc #14770: [META] DevelopersMeeting added

#2 - 09/16/2018 01:28 PM - mame (Yusuke Endoh)
  - Description updated

#3 - 09/16/2018 02:19 PM - mame (Yusuke Endoh)
  - Description updated

#4 - 09/16/2018 02:29 PM - mame (Yusuke Endoh)
  - Description updated

#5 - 09/18/2018 04:24 PM - greggzst (Grzegorz Jakubiak)
  - [Feature #15123] Enumerable#compact proposal (greggzst)
    It simplifies working with large and small collections so one doesn't have to remember that can't use #compact when enumerator is returned and have to fall back to #reject(:nil?).

08/27/2022
# Feature suggestion #15112 Introduce the new singleton method STDERR.p (by Kenta Murata)

I am mostly curious what the ruby core team thinks about Kenta Murata’s proposal; it probably will not take too much time away discussing it briefly, since the scope is small.

# Feature #11505 Module#=== should call #kind_of? on the object rather than rb_obj_is_kind_of which only searches the ancestor hierarchy
   ○ This would allow patterns as Decorator and Proxy to work with case statements.

# Feature #14912 Introduce pattern matching syntax (greggzst)

Many modern languages have introduced pattern matching. I used it in scala and found it very easy to utilize and understand especially in recursion. It makes extracting data easier as well.

# Feature #15144 Enumerator#chain

# Feature #14799 Startless range: usefulness discussed, patch provided by @mame (Yusuke Endoh), waits for Matz's decision (?)
# Feature #14784 Comparable#clamp accepting range: comment for @akr (Akira Tanaka) about "needlessly big" proposal, I answered it, is it makes the proposal more likely to be accepted?
# Feature #5284 Composition for procs: the last thing Matz has said is the operators are chosen (<< and >>), and "We need more discussion if we would add combination methods to the Symbol class." Is there a chance proc composition would make it way in the 2.6?
# Feature #13958 Syntax sugar for method reference. The last thing Matz have said is: ":-: looks best to me (followed by ::). Let me consider it for a while." Are there any choices made? Could we expect this for 2.6?
# Feature #14781 Enumerator#generate It seems like people feel cautious enthusiasm about it, but not sure about the name. What should be done here? Voting on the name? Providing the patch with some name, and then voting for the name?

# Feature #15195 How to deal with new Japanese era
   ○ We should prepare early (even if it's just to check that we need to do nothing)

# Feature #15107 How to deal with capitalizing Georgian in Unicode 11.0.0

I need feedback on this to be able to implement in time for the Ruby 2.6 release.

# Feature #15144 Enumerator#chain

# Feature #14799 Startless range: usefulness discussed, patch provided by @mame (Yusuke Endoh), waits for Matz's decision (?)
# Feature #14784 Comparable#clamp accepting range: comment for @akr (Akira Tanaka) about "needlessly big" proposal, I answered it, is it makes the proposal more likely to be accepted?
# Feature #5284 Composition for procs: the last thing Matz has said is the operators are chosen (<< and >>), and "We need more discussion if we would add combination methods to the Symbol class." Is there a chance proc composition would make it way in the 2.6?
# Feature #13958 Syntax sugar for method reference. The last thing Matz have said is: ":-: looks best to me (followed by ::). Let me consider it for a while." Are there any choices made? Could we expect this for 2.6?
# Feature #14781 Enumerator#generate It seems like people feel cautious enthusiasm about it, but not sure about the name. What should be done here? Voting on the name? Providing the patch with some name, and then voting for the name?

# Feature #14799 Startless range: usefulness discussed, patch provided by @mame (Yusuke Endoh), waits for Matz's decision (?)
# Feature #14784 Comparable#clamp accepting range: comment for @akr (Akira Tanaka) about "needlessly big" proposal, I answered it, is it makes the proposal more likely to be accepted?
# Feature #5284 Composition for procs: the last thing Matz has said is the operators are chosen (<< and >>), and "We need more discussion if we would add combination methods to the Symbol class." Is there a chance proc composition would make it way in the 2.6?
# Feature #13958 Syntax sugar for method reference. The last thing Matz have said is: ":-: looks best to me (followed by ::). Let me consider it for a while." Are there any choices made? Could we expect this for 2.6?
# Feature #14781 Enumerator#generate It seems like people feel cautious enthusiasm about it, but not sure about the name. What should be done here? Voting on the name? Providing the patch with some name, and then voting for the name?