## Status
- **Closed**
- **Normal**

### Description
Please comment on 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 the ticket comments)

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

Date: 2019/12/20 13:00-17:00
Place, Sign-up, and Log: [https://docs.google.com/document/d/18v0UWXwwKasvOFp1UQbEnt_4p0BMTdoGgcNfjyAWTKs](https://docs.google.com/document/d/18v0UWXwwKasvOFp1UQbEnt_4p0BMTdoGgcNfjyAWTKs)

### Agenda
- Next dev-meeting
- About 2.7 timeframe
- Check security tickets
- Discussion

Please comment on 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 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 this feature.

Please follow the comment format strictly! We'll use this script to automatically create an markdown-style agenda. We may ignore a comment that violates the format.

A short summary of a ticket is strongly recommended. We cannot read all discussion of the ticket in a limited time.

A proposal is often changed during the discussion, so it is very helpful to summarize the latest/current proposal, post it as a comment in the ticket, and write a link to the comment.

### Related issues:
- Related to Ruby master - Misc #14770: [META] DevelopersMeeting

**Open**
History

#1 - 12/03/2019 10:33 AM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 12/03/2019 10:34 AM - mame (Yusuke Endoh)

Carry over:

- [Feature #16264] Real "callable instance method" object. ..:method to be a first-class thing, instead of Symbol#to_proc trick (zverok)
  - This proposal would make sense only if ..: would not be reverted (which, to the best of my understanding, is now doubtful)

- [Misc #16291] Introduce support for resize in rb_ary_freeze and prefer internal use of rb_ary_freeze and rb_str_freeze for String and Array types (lourens)
  - Builds onto the capacity shrinking feature introduced by rb_str_freeze, targeting Array
  - Many internal uses that froze String types did not use the rb_str_freeze variation and could not benefit from resizing capacity on freeze
  - Implemented the same for Array
  - Let Array#freeze call rb_ary_freeze to expose the shrinking capability to user code too (as recommended by Shyouhei) for parity with String#freeze already doing the same

- [Bug #15620] Block argument usage affects lambda semantic (alanwu)
  - I find the current behaviour unreasonably confusing and would like to see improvement, even though the bug doesn't really show up in the real world often.
  - I have a pull request which restores the behaviour found in Ruby 2.4.x and warns about it, based off of matz's comment in [ruby-core:94054].
  - Could someone confirm that is the desired fix? If it is the fix we want, could someone review the PR?

- [Feature #16274] Transform hash keys by a hash (sawa)
  - Easily rename hash keys that do not follow a rule

- [Misc #16375] Right size regular expression compile buffers for literal regexes and on Regexp#freeze (lourens)
  - Builds on Misc #16291, I think there's potential to apply this pattern to other types at hooks outlined at the end of the issue
  - A large set of literal regular expressions are quite common in Rails applications (mostly framework, but also application and dependencies)
  - In my Redmine boot test was able to shave 300kb off just excess regex buffer capacity

- [Misc #16260] Symbol#to_proc behaves like lambda (zverok)
  - (dummy comment)

#3 - 12/03/2019 10:40 AM - mame (Yusuke Endoh)

- [Bug #6087] How should inherited methods deal with return values of their own subclass? (mame)
  - Matz said "We will fix this (to consistently return Arrays) in 3.0." seven years ago. Now is the time. Final confirmation.

#4 - 12/03/2019 11:46 AM - mame (Yusuke Endoh)
- Description updated

#5 - 12/03/2019 12:03 PM - mame (Yusuke Endoh)
- Description updated

#6 - 12/03/2019 12:04 PM - mame (Yusuke Endoh)

NOTE: Please follow the comment format strictly!

I've created a script to automatically create an markdown-style agenda.
https://gist.github.com/mame/b0390509ca1491b43610b9eb665eb86

We may ignore a comment that violates the format.

#7 - 12/03/2019 11:59 PM - jeremyevans0 (Jeremy Evans)


- **Feature #14183** "Real" keyword argument (jeremyevans0)
  - Is it OK to merge branch to remove deprecated support for positional hash <-> keyword conversion after 2.7 released?

- **Bug #11022** opening an eigenclass does not change the class variable definition context (jeremyevans0)
  - Discussed during September dev meeting, matz wanted to consider for a while. Has a decision been made?

- **Bug #7844** include/prepend satisfiable module dependencies are not satisfied (jeremyevans0)
  - Discussed during September dev meeting, matz wanted to consider for a while. Has a decision been made?

- **Bug #14249** warn four special variables: $; $; $; $ (jeremyevans0)
  - Do we still want to warn regarding these variables? If so, should the warnings be during parsing or at runtime (if variables are aliased and then modified)?

- **Feature #10463** $@ and :$@ are not parsed correctly (jeremyevans0)
  - matz decided in July that def $@ and def $@ should continue to work. However, can we fix the parser to not treat $@ and :$@ as $@ and $@?

#8 - 12/11/2019 10:39 AM - byroot (Jean Boussier)

- **Feature #16377** Regexp literals should be frozen (byroot)
  - Regex literals always reference the same mutable instance. This allow to leak global state with instance_variable_set
  - Change already accepted by Matz about 2 years ago: https://bugs.ruby-lang.org/issues/8948#note-14, but then nothing happened.

#9 - 12/16/2019 04:43 PM - alanwu (Alan Wu)

- **Bug #16260** Symbol#to_proc behaves like lambda, but doesn't acknowledge it (alanwu)
  - This should make Proc#lambda? more accurate.

- **Bug #16406** (lambda_proc << normal_proc).lambda? should return false (alanwu)
  - I think this is more intuitive than the current behavior.

#10 - 12/18/2019 12:07 AM - znz (Kazuhiro NISHIYAMA)

- **Feature #16419** FrozenError.new ignores receiver: (znz)
  - It seems that no one have strong opinion. So I want matz to decide.

- **Feature #16420** Warning[experimental]=false (znz)
  - I heard this more intuitive than the current behavior.

#11 - 12/19/2019 02:04 AM - mame (Yusuke Endoh)

- **Bug #9815** attr_reader doesn't warn on a uninitialized instance variable (mame)
  - A reader method defined by attr_reader :foo is not warned as "instance variable @foo not initialized". Is it intentional?

#12 - 12/20/2019 01:02 AM - mame (Yusuke Endoh)

- **Bug #8841** Module#included_modules and prepended modules (mame)
  - Module#include? and Module#included_modules regard prepended modules as included (not well documented); Module#included is not called when the module is prepended. Is this right?
  - IMO, changing the behavior is no longer acceptable (without any actual trouble). How about just changing the document?

- **Feature #8026** Need Module#prepended_modules (mame)
  - It is accepted six years ago, but not implemented yet. I've never heard any actual trouble, but should we still add the feature?

#13 - 12/20/2019 01:06 AM - mame (Yusuke Endoh)

- **Bug #9815** attr_reader doesn't warn on a uninitialized instance variable (mame)
  - A reader method defined by attr_reader :foo is not warned as "instance variable @foo not initialized". Is it intentional?

#14 - 12/20/2019 01:22 AM - mame (Yusuke Endoh)

- **Bug #10388** Operator precedence problem in multiple assignment (massign) (mame)
  - "a, b = c = 1, 2 is currently taken as a, b = (c = 1), 2; I'd expect it to be taken as a, b = (c = 1, 2)." Jeremy gave a try to implement but seemed difficult due to the limitation of LALR(1) parser. Let's give up.
- [Bug #10475] Array#flatten should not accept a nil argument (mame)
  - Should we add a document that Array#flatten accepts nil? Negative argument too?

#15 - 12/20/2019 01:29 AM - mame (Yusuke Endoh)
[Bug #10929] NilClass#to_proc and & don't mix? (mame)
  - I think it is not worth adding.

#16 - 12/20/2019 01:39 AM - mame (Yusuke Endoh)
  - [Bug #11014] String#partition doesn't return correct result on zero-width match (mame)
    - I'd like to confirm if the current behavior is intended.

#17 - 12/20/2019 02:15 AM - soutaro (Soutaro Matsumoto)
I'd like to discuss about type signature integration for Ruby3. I plan to discuss in this meeting but not trying to make some decisions today. Asking for feedbacks to make a concrete proposal for the next meeting or else.

A doc is available to introduce what we have been doing.
https://docs.google.com/document/d/1QVma3sr4GkGqL37yPEK5LoWRzl09UJaOL-vh-vqUQ/edit#

#18 - 12/20/2019 02:22 AM - osyo (manga osyo)
  - [Feature #16432] Using _1 inside binding.ire in a block that uses _1 and using _1 in irb will cause unintended behavior (osyo)
    - Calling binding.ire in a block that uses _1 and using _1 in irb will cause unintended behavior.
    - Should it be a runtime error?

#19 - 12/26/2019 07:15 AM - mame (Yusuke Endoh)
- Status changed from Open to Closed