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: 2019/07/11 (Thu)
Time: 13:00-17:00 (JST)
Place and Sign-up: https://ruby.connpass.com/event/135823/
log: https://docs.google.com/document/d/1K61SGIwp8_rNpPyhmayUcERu71vtedjXdhqrLmBVY/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 the 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 are scheduled according to when/where we can reserve Matz's time.

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 (kol)
  * I feel this feature is very useful and some people say :+1: so let discuss this feature.

We don't guarantee to put tickets in agenda if the comment violate the format (because it is hard to copy&pase).

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

History

#1 - 06/17/2019 03:20 AM - mame (Yusuke Endoh)

Carry-over tickets.
Could the pattern matching be made available as a first-class citizen to be used as a filter when searching in data structures and to be able to implement Actor receive method
- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

Could the pattern matching be made available as a first-class citizen to be used as a filter when searching in data structures and to be able to implement Actor receive method
- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

Could the pattern matching be made available as a first-class citizen to be used as a filter when searching in data structures and to be able to implement Actor receive method
- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)

Could the pattern matching be made available as a first-class citizen to be used as a filter when searching in data structures and to be able to implement Actor receive method
- Consider contractions log_messages.find in [:error, message] { puts message }
- Non-symbol matching of Hashes would be very desirable
- Elaborated in [https://bugs.ruby-lang.org/issues/14912#note-22](https://bugs.ruby-lang.org/issues/14912#note-22)
What's the reason of current behavior? Please explain on the ticket.
matz: Can you judge whether that behavior is worth the inconsistency with `{ |x|` and making `#map`, etc not work with unknown element types (if elements can or not be arrays)?

#11 - 06/30/2019 12:14 AM - nobu (Nobuyoshi Nakada)

- [Feature #11808] Different behavior between Enumerable#grep and Array#grep
  It needs a new C-API.

#12 - 06/30/2019 12:25 PM - Eregon (Benoit Daloze)

- [Feature #15966] Introducing experimental features behind a flag, disabled by default
  - Could you read the description and reply what you think? Do you think it's a good idea? I think it would be a much better way to introduce experimental features.

#13 - 07/02/2019 10:07 AM - byroot (Jean Boussier)

- [Feature #15940] Coerce symbols internal fstrings in UTF8 rather than ASCII to better share memory with string literals
  - Saves some resident memory
  - Makes symbols & constant names defined in UTF-8 files UTF-8 encoded, which is much less surprising.
  - Little to no backward compatibility concerns.
- [Feature #15939] Dump symbols reference to their fstr in ObjectSpace.dump()
  - It's important for the heap dump consistency. Otherwise when you build the reference graph you end up with some dangling objects.

#14 - 07/02/2019 01:29 PM - nobu (Nobuyoshi Nakada)

- [Feature #15973] Make Kernel#lambda always return lambda

#15 - 07/04/2019 09:01 AM - nobu (Nobuyoshi Nakada)

- [Feature #11460] Unhelpful error message when naming a module with the same name as an existing class

#16 - 07/05/2019 01:24 PM - methodmissing (Lourens Naudé)

- [Feature #15631] Let round_capa for ID table not allocate excess capacity for power of 2 ints >= 4
  - PR and further due diligence comments in https://github.com/ruby/ruby/pull/2278

#17 - 07/06/2019 03:36 PM - Eregon (Benoit Daloze)

- [Feature #15987] Let boolean option (such as exception in Kernel#Complex, Kernel#Float, Kernel#Integer, Kernel#Rational) be falsy vs. truthy
  - Does this issue need matz approval or can it be fixed directly? If it does, does matz agree we should address this and use RTEST()?

#18 - 07/10/2019 04:33 AM - mame (Yusuke Endoh)

- [Feature #15865] `<expr>` in `<pattern>` expression (mame)
  - We have some opinions: scoping, matching strictness, the keyword in, and the word order. But all are not specific to the one-line matching. I think it is acceptable if case/in matching is acceptable. May I experimentally commit it as well as case/in?

#19 - 07/10/2019 04:48 AM - jeremyevans0 (Jeremy Evans)

- [Bug #10463] `::@` and `!:@` are not parsed correctly (jeremyevans0)
  - Can we deprecate the automatic conversion of `~@` to `~` and `!@` to `!` in method names and symbols?

#20 - 07/11/2019 09:23 AM - methodmissing (Lourens Naudé)
- [Misc #15805] Let memory sizes of the various IMEMO object types be reflected correctly
  - Only the imemo_tmpbuf type’s auxiliary malloc heap buffer is factored into obj_memsize_of, other imemo types also alloc on the heap
  - PR: https://github.com/ruby/ruby/pull/2140
- [Misc #15806] Explicitly initialise encodings on init to remove branches on encoding lookup
  - Currently rb_enc_from_index and related methods explicitly check for encoding table init on each call - I think this should be initialised at boot time instead as even the simple case of loading the interpreter requires the encoding table to be loaded anyways (symbol.c)
  - PR: https://github.com/ruby/ruby/pull/2128

#21 - 07/11/2019 11:34 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed