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 cannot 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 (ko1)
  * 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 the agenda if the comment violates the format (because it is hard to copy&paste).

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

- Related to Misc #14770: [META] DevelopersMeeting added

- Description updated

#3 - 09/08/2019 03:49 PM - jeremyevans0 (Jeremy Evans)
- [Bug #12984] rescue *[] should be equivalent to rescue as method_call([]) is equivalent to method_call
  - I think the current behavior is expected. Can matz confirm?
- [Bug #11022] opening an eigenclass does not change the class variable definition context
  - I think the current behavior is expected. Can matz confirm?
- [Bug #8297] extend & inherited class variable issue
  - I would like to fix Module#class_variables to be consistent with Module#class_variable_get using the patch.
- [Bug #7844] include/prepend satisfiable module dependencies are not satisfied
  - I would like to make it so Module#include does not prepend modules before the receiver using the patch.
- [Bug #7522] Non-core "Type()" Kernel methods return new object
  - I would like to make it so Kernel#{BigDecimal,Complex,Pathname} return argument if given correct type using the patch.
- [Bug #11536] super in instance_eval in a method defined in a module is invoked with a wrong receiver
  - I would like to raise an exception instead, similar to super in instance_eval in method defined in class using the patch.

#4 - 09/08/2019 08:09 PM - Eregon (Benoit Daloze)
- [Feature #16150] Add a way to request a frozen string from to_s
  - Can I commit https://github.com/ruby/ruby/pull/2437 (make Symbol#to_s frozen)? It seems compatible enough. What about making true/false/nil#to_s return a frozen cached String?

#5 - 09/09/2019 04:13 AM - mame (Yukske Endoh)
- [Bug #16154] lib/delegate.rb issues keyword argument warnings
  - Module#pass_positional_hash for disabling false positive warnings in delegation. This is just for 2.7 migration path. Can we commit this?

#6 - 09/09/2019 09:18 PM - methodmissing (Lourens Naudé)
- [Misc #16124] Let the transient heap belong to objspace (methodmissing)
  - As per PR and outline of pros and cons in the issue, I think it's stable and performance enough to be coupled a little tighter to objspace instead of just being a global slab / scratch space. Thoughts?
  - I volunteer to do the work, if it makes sense.
  - Also VM -> global method cache for example - cleaner for multi-VM, but not sure of the future of MVM

#7 - 09/09/2019 09:40 PM - methodmissing (Lourens Naudé)
- [Misc #16160] Lazy init thread local storage (methodmissing)
  - Storage for Fiber local storage of execution context for Thread#[] and Thread#[]= APIs are lazy initialized.
  - I think the same pattern should be applied to real thread local storage as well.

#8 - 09/11/2019 03:03 PM - alanwu (Alan Wu)
- [Feature #16131] Remove $SAFE, taint and trust
  - What do you think about the proposed schedule?
  - I am asking because this might inform the fix for [Bug #16151]
- [Feature #16123] Allow calling a private method with self.
  - Ping because of inactivity. I think it makes sense.

Patches that would benefit from reviews:
- [Bug #16121] Stop making a redundant hash copy in Hash#dup
- [Feature #16163] Reduce the output of RubyVM::InstructionSequence#to_binary
- [Bug #16119] Optimize Array#flatten and flatten! for already flattened arrays

#9 - 09/15/2019 05:58 PM - connorshea (Connor Shea)
- [Feature #16155] Add an Array#intersection method
  - Is this an acceptable method name?
  - Should this be a simple alias for Array#&, or should it accept multiple arrays as arguments, like difference and union do?

#10 - 09/15/2019 09:02 PM - jeremyevans0 (Jeremy Evans)

- [Feature #16168] Add keyword argument separation to C functions using rb_scan_args
  - I would like to commit the patch so that methods implemented in C handle keyword argument separation similar to methods implemented in Ruby.

#11 - 09/18/2019 11:14 AM - hsbt (Hiroshi SHIBATA)

[Feature #16170] Remove the unmaintained libraries from Ruby 2.7

- Is it ok to remove them?

#12 - 09/18/2019 02:11 PM - Dan0042 (Daniel DeLorme)

- [Feature #16120] Omitted block argument if block starts with dot-method call
  - posts.map{ .author.name } has beautiful readability, no incompatibility, changes a syntax error into an intuitive idiom, already implemented in 6 lines by nobu. 本当に美しいと思います。

- [Feature #16153] eventually_frozen flag to gradually phase-in frozen strings
  - This would make previous backward-incompatible proposals (#16150, #15836, #11473) possible with a backward-compatible transition.

- [Feature #16021] floor/ceil/round/truncate should accept a :step argument
  - Since Time#floor/ceil methods were added, it would be convenient to round to the previous/next minute/hour: Time.now.ceil(step:60). Also add to Numeric, for convenience and consistency.

  - If this is a good idea I would like to try writing it.

#13 - 09/19/2019 01:55 AM - shyouhei (Shyouhei Urabe)

[Misc #16172] C++ situation?

- What do we do with C++ extensions?

#14 - 09/20/2019 03:20 AM - shyouhei (Shyouhei Urabe)

Can someone update the "Log" link above?

#15 - 09/21/2019 07:56 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

I published the Log. Closing.