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

- [Feature #15554] warn/error passing a block to a method which never use a block (ko1)
  - I want to share the ticket and survey on some product.

From non-attendees

- [Feature #6470] Make attr_accessor return the list of generated method (eregon)
  - I think not all related issues were considered. There are use cases, both for private attr_* and other modifiers (e.g., volatile, atomic, tracing accesses, getter/setter decorators, etc).
- [Bug #14353] $SAFE should stay at least thread-local for compatibility (eregon)
  - I would like matz's opinion on this. It seems the compatibility risks and debugging problems is not worth making $SAFE global, unless there is a good reason to be global? Should there still be $SAFE checks at all, for e.g., tainted Strings?
- [Feature #5653] "I strongly discourage the use of autoload in any standard libraries" (eregon)
  - It would be useful to reach a decision for this. What does it gain, and at what cost for compatibility? Is there going to be a replacement?
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 (kol)
  * I feel this feature is very useful and some people say :+1: so let's discuss about this feature.

We 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

History
#1 - 01/19/2019 12:22 AM - shyouhei (Shyouhei Urabe)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 01/19/2019 05:24 AM - aycabta (aycabta .)
- [Feature #4475] default variable name for parameter (aycabta)
  * In #15483, matz said "I feel the expression ary.map(&{:to_i << :chr}) is far less readable than ary.map{|x|x.to_i.chr}"., and "And this can lead to the default block parameter like it."

#3 - 01/19/2019 06:05 AM - mame (Yusuke Endoh)
- Description updated
- Subject changed from DevelopersMeeting201900207Japan to DevelopersMeeting20190207Japan

#4 - 01/19/2019 12:51 PM - Eregon (Benoit Daloze)
- Description updated

#5 - 01/19/2019 12:51 PM - Eregon (Benoit Daloze)
- Description updated

#6 - 01/21/2019 03:46 AM - hsbt (Hiroshi SHIBATA)

The ancient issues that is no response. I'm okay to discuss them with our spare time of this meeting.

- [Feature #11796] [PATCH] Refactor reduce call get_stat(): nobu?
- [Feature #11710] [PATCH] Replace Set#merge with Set#merge! and make Set#merge non-mutating.: knu?
- [Feature #10782] Patch: Add constants for BigDecimal for ZERO, ONE, TEN: mrkn?
- [Feature #10459] [PATCH] rfc3339 method for Time: akr?
- [Feature #9556] Add HTTP#get block functionality to HTTP.get: naruse?
- [Bug #11273] [PATCH] Make it possible to load from a FIFO file: akr?
- [Bug #11304] [PATCH] Kernel.global_variables should observe $~.
- [Misc #11373] Add command line option to query valid values for --dump
- [Bug #11391] chopped_length should return early for empty strings
- [Feature #10175] There's no reason to prefer Proc.new over Kernel#proc anymore
- [Feature #10370] [PATCH] We don't need to check whether rb_block_call exists
- [Feature #8184] Avoid the creation of meaningless ranges (nil, false, true)

#7 - 01/21/2019 09:30 AM - mame (Yusuke Endoh)

- [Feature #5632] Attempt to open included class shades it instead. (mame)
  * Matz, do you still want to "fix" this issue? And how?

#8 - 01/23/2019 10:45 AM - Eregon (Benoit Daloze)
- Description updated

#9 - 01/23/2019 10:57 AM - Eregon (Benoit Daloze)
• [Feature #11076] Enumerable method count_by
  A new name "tally" is proposed.  "Tally" is: | → || → ||| → |||| → |||||.
  Looks good to me.  What do you think?

• [Feature #14344] refine at class level (kddeisz)
  Shorter refinement syntax proposed based on common use cases. Proposed either refine_class or using blocks for shorthand. What do you think?
• [Feature #14680] Adding +@ and -@ to hash and array (kddeisz)
  Since we have -@ and +@ for strings and it's very useful I'd like to propose adding the same API to hash and array.

• [Feature #15567] Allow ensure to match specific situations. (ioquatix)
  Could we discuss whether this is a good idea or not?
• [Feature #15560] Add support for read/write offsets. (ioquatix)
  Can we get consensus on whether we can add this?

• [Feature #15573] Permit zero step in Numeric#step and Range#step
  I make this issue as a feature request because the behaviors exist at least since version 2.3. But I think it is a bug, so it should be resolved.

• [Feature #15581] Split tool/* files to tool and script directories
  What's a concern about this?

• [Feature #10098] Timing-safe string comparison for OpenSSL::HMAC (bdewater)
  The feature is done (the title is old, it will be implemented in String) but blocked on picking a method name. I would like to ask for opinions so this can move forward.

• [Feature #15331] [PATCH] Faster hashing for short string literals
  Looking for feedback on this patch.

• [Feature #15374] Proposal: Enable refinements to #method_missing
  waited one month.

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

• [Feature #15588] String#each_chunk and #chunks
  Looking for feedback on this patch.
Status changed from Open to Closed