Ruby - Misc #17041

DevelopersMeeting20200831Japan

07/22/2020 05:55 AM - mame (Yusuke Endoh)

Status:	Closed	
Priority:	Normal	
Assignee:		

Description

The next dev meeting

Date: 2020/08/26 13:00-17:00
Place/Sign-up/Agenda/Log: https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20200826Japan.md

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

Call for agenda items

If you have a ticket that you want matz and committers to discuss, please post it into this ticket in the following format:

```
* [Ticket ref] Ticket title (your name)
  * Comment (A summary of the ticket, why you put this ticket here, what point should be discussed
, etc.)
```

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.
```

- Comment deadline: 2020/08/19 (one week before the meeting)
- The format is strict. We'll use this script to automatically create an markdown-style agenda. We may ignore a comment that does not follow the format.
- Your comment is mandatory. We cannot read all discussion of the ticket in a limited time.

Related issues:

Related to Ruby - Misc #14770: [META] DevelopersMeeting	**Open**

History

#1 - 07/22/2020 05:55 AM - mame (Yusuke Endoh)

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

#2 - 07/22/2020 05:58 AM - znz (Kazuhiro NISHIYAMA)

- [Feature #17039] Remove Time#succ (znz)
  - Time#succ is obsolete since 1.9.2.

#3 - 07/22/2020 06:06 AM - shyouhei (Shyouhei Urabe)

- [Feature #17040] cleanup include/ruby/backward* (shyouhei)
  - Is anybody against it?

#4 - 07/22/2020 08:04 AM - mame (Yusuke Endoh)

Please note that it may not be possible to go though all topics that are raised in this agenda because we will discuss 2.8/3.0 release schedule. (Perhaps, we will prioritize the topics related to 2.8/3.0, such as Ractor.) Thanks.

#5 - 07/22/2020 05:28 PM - marcandre (Marc-Andre Lafortune)

- [Feature #16989] Sets need ♥️, aka the "Set Program" (marcandre)
  - Bring Set into core
  - Insure interoperability with Array (e.g so array & set works and is efficient)
  - Shorthand syntax for static frozen sets of string/symbols (e.g. %ws{hello world})

#6 - 07/28/2020 10:07 PM - jeremyevans0 (Jeremy Evans)

- [Feature #17055] Allow suppressing uninitialized instance variable and method redefined verbose mode warnings (jeremyevans0)
  - This keeps the default behavior the same as before, but allows opt-in to supress the warnings.
  - This allows for high-performance code (uninitialized instance variables) and safe code (redefining methods without removing in multi-thread environments) to work without issuing warnings in verbose mode.
  - Is it OK to commit the patch?

#7 - 08/01/2020 01:27 PM - Eregon (Benoit Daloze)

- [Feature #14844] Future of RubyVM::AST? (eregon)
  - Other Ruby implementations need to be able to implement it too given more and more gems use it. Where do we move it?
- [Feature #15752] A dedicated module for experimental features (eregon)
  - Create ExperimentalFeatures and move things there or RubyVM exists on all Ruby implementations? Please choose.

#8 - 08/03/2020 03:55 PM - Dan0042 (Daniel DeLorme)

- [Bug #17017] Range#max & Range#minmax incorrectly use Float end as max (dan0042)
  - (1..3.1).max has resulted in 3.1 since ruby 1.9 and has now been "fixed" to return 3. Matz, please confirm?

#9 - 08/04/2020 08:32 AM - byroot (Jean Boussier)

- [Feature #17103] Add a :since option to ObjectSpace.dump_all (byroot)
  - On large heap, dump_all can take over a minute and generate huge (6+GiB) files.
  - since: 42, means only objects matching generation && generation >= 42 would be dumped.
  - Sometimes you are only interested in objects allocated past a certain point, this make that use case much faster and efficient.

#10 - 08/06/2020 10:14 AM - Eregon (Benoit Daloze)

- [Feature #17104] Do not freeze interpolated strings when using frozen-string-literal (eregon)
  - It seems many people agree there is no point to freeze interpolated strings.
  - Interpolated strings are not "simple literals" (just like 1 + 2 is not a literal) and they involve mutation anyway, it seems of very little value to freeze, and it's inconvenient.
  - @bughit: The reasons are, it will reduce allocations, be more logical, less surprising and produce simpler code (when a mutable string is needed and you don't want extra allocations)

#11 - 08/09/2020 08:19 AM - byroot (Jean Boussier)

- [Feature #13381] Expose rb_fstring and its family to C extensions
  - It's only missing a decision on the names. Could this be discussed again?

#12 - 08/09/2020 06:53 PM - Eregon (Benoit Daloze)

- [Bug #17105] A single return can return to two different places in a proc inside a lambda inside a method (eregon)
  - Is current implementation correct? Should it not be a LocalJumpError if the surrounding lambda is no longer on stack?

#13 - 08/19/2020 05:15 AM - znz (Kazuhiro NISHIYAMA)

- [Feature #17125] Remove Thread.exclusive (znz)
  - Thread.exclusive is deprecated since 2.3.

#14 - 08/19/2020 04:26 PM - mame (Yusuke Endoh)

- [Feature #17122] Add category to Warning#warn (mame)
  - I'd like to hear opinions from the committers

#15 - 08/25/2020 10:39 PM - mrkn (Kenta Murata)

- [Feature #14723] python's buffer protocol clone (mrkn)
    - I renamed buffer to memory_view.
        - This name is already used in Python for the Python object wrapper of PyBuffer.
    - I've added tests for some functions and a simple memory view implementation of a String for testing these functions.
        - https://github.com/ruby/ruby/pull/3261/files

#16 - 08/26/2020 01:16 AM - mrkn (Kenta Murata)

- [Feature #16812] Allow slicing arrays with ArithmeticSequence (mrkn)

    - I found the wrong test case, so the implementation was also in wrong state. The proposed patch has been fixed.

    - There are opposing opinions in the following case:

      ```
      [0,1,2,3,4,5][(6..0)%-2] #=> [5, 3, 1]
      ```

      ```
      # or
      ```

      ```
      [0,1,2,3,4,5][(6..0)%-2] #=> [4, 2, 0]
      ```

    - Python follows the former.

      ```
      >>> [0, 1, 2, 3, 4, 5][6::-2]
      [5, 3, 1]
      ```

#17 - 08/26/2020 08:40 AM - mame (Yusuke Endoh)

The meeting was postponed to the next Monday (31th Aug.) for some reason.

#18 - 08/27/2020 07:43 AM - duerst (Martin Dürst)

- Subject changed from DevelopersMeeting20200826Japan to DevelopersMeeting20200831Japan

Adjust subject to match new date.

#19 - 09/01/2020 03:47 PM - mame (Yusuke Endoh)

- Description updated

- Status changed from Open to Closed