The next dev meeting

Date: 2021/04/16 13:00-17:00
Place/Sign-up/Agenda/Log: https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20210416Japan.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 (kol)
  * I feel this feature is very useful and some people say :+1: so let discuss this feature.

It is recommended to add a comment by 2021/04/13. We hold a preparatory meeting to create an agenda a few days before the dev-meeting.

The format is strict. We'll use this script to automatically create a 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. We appreciate it if you could write a short summary and update from a previous discussion.

Related issues:
Related to Ruby master - Misc #14770: [META] DevelopersMeeting Open
I don't think we can work around that without keeping the abstract syntax tree or source code of all methods in memory, which seems wasteful.

- Is the current behavior a bug, or should we just accept the current behavior as spec and close this?

- [Bug #9542] Delegator does not delegate protected methods (jeremyevans0)
  - Do we want to add a way to tell if a method was called with an implicit receiver or self?
  - We may be able to implement this via a VM frame flag and a method to check it.
  - If we don't want to add such a way, can we accept the current behavior as spec and close this?

- [Bug #11230] Should rb_struct_s_members() be public API? (jeremyevans0)
  - This is still public in the C-API.
  - Do we want to keep it and close this issue, or do we want to remove it?
  - If we want to remove it, how should it be deprecated?

#3 - 04/02/2021 11:57 PM - tenderlovemaking (Aaron Patterson)

- [Feature #17749] Add Module#source_location
  - It's similar to the const_source_location feature, but easier to use when you don't know the constant name, or enclosing const
  - This is useful for debugging

- [Feature #17753] Add Module#namespace
  - A::B.namespace will return A
  - This is to help with finding "sibling" constants of B so we don't have to parse the constant name
  - This is also useful for debugging

#4 - 04/07/2021 06:33 AM - mame (Yusuke Endoh)

- [Feature #17762] A simple way to trace object allocation
  - I want require "objcpace/trace" or something which is a very useful debugging tool to identify the allocation site of an object.

#5 - 04/07/2021 04:24 PM - Dan0042 (Daniel DeLorme)

- [Feature #17692] String#casecmp performance improvement (dan0042)
  - Is it ok to make casecmp faster?

#6 - 04/08/2021 12:35 PM - marcandre (Marc-Andre Lafortune)

- [Feature #15198] Add Array#intersect? (marcandre)
  - Ok to add?

#7 - 04/08/2021 01:05 PM - byroot (Jean Boussier)

- [Feature #17795] before_fork and after_fork callback API
  - Many libraries out there use various tricks to detect the process being forked, most commonly Process.pid != @pid, but glibc no longer cache the PID so these libraries end up doing syscalls in tight loops.
  - If a callback API is too complicated, I proposed a simple alternative that is to have Kernel.fork be a simple delegator to Process.fork
    https://github.com/ruby/ruby/pull/4361. This would makes it easy for libraries to decorate fork to have their own callbacks on either before or after fork.

#8 - 04/13/2021 03:07 PM - parker (Parker Finch)

- [Feature #17016] Add Enumerable#accumulate (parkerfinch)
  - Is the name #accumulate acceptable?
  - Is this feature ok to add?

#9 - 04/17/2021 08:18 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

- Description updated