The next dev meeting

Date: 2022/01/13 13:00-17:00 (JST)

- 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 record of the discussion in the 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 of the meeting 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.

- It is recommended to add a comment by 2022/01/10. 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

History

#1 - 12/09/2021 12:31 PM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 12/09/2021 01:27 PM - zverok (Victor Shepelev)

  - [Feature #18368] Range#step semantics for non-Numeric ranges (zverok)
    * Is there a reason why (timestamp1...timestamp2).step(3.hours) is not what #step supports?
    * If there is, can there be a new method introduced with this semantic?
  - [Feature #18136] Enumerable#take_while_after (zverok)
    * Many real-life use-cases provided
    * Two alternative names provided: take_while_after and drop_after
    * Is there any way this can be moved forward?

#3 - 12/09/2021 03:45 PM - mame (Yusuke Endoh)
Stop the interpreter from escaping error messages (mame)

> At the previous dev-meeting, we decided to survey the current status of how modern terminal emulators handle untrusted escape sequences.
> I surveyed rxvt, Eterm, xterm, VTE (gnome-terminal), Windows Terminal, and item2, and all of them have addressed known vulnerabilities.
> Also, according to some comments in source code, terminal authors are completely aware that this kind of vulnerabilities should be addressed on terminal emulator side.
> I believe there is already a consensus that this kind of security issues should be handled by terminal emulators, not Ruby.
> So, let's remove the escape!

#4 - 12/25/2021 12:18 PM - mame (Yusuke Endoh)

Notes on issues found during the release process: https://hackmd.io/qUbiOiyRQXibCfLRbzS9Og (Sorry for Japanese)

#5 - 01/02/2022 01:29 AM - duerst (Martin Dürst)

[Bug #18294] error when parsing regexp comment (Martin Dürst)
- We should decide whether this is spec, or to make parsing easier, or a bug.

#6 - 01/03/2022 06:22 PM - jeremyevans0 (Jeremy Evans)

[Bug #15928] Constant declaration does not conform to JIS 3017:2013 (jeremyevans0)
- This was discussed at the May 2021 developer meeting, and it was decided to wait until after 3.1.
- This makes constant assignment consistent with attribute assignment, so evaluation occurs left-to-right.
- Is this OK to commit now?
[Bug #17545] Calling dup on a subclass of Proc returns a Proc and not the subclass (jeremyevans0)
- Is this OK to commit this now?
[Bug #16633] Add block or filtered forms of Kernel#caller to allow early bail-out (jeremyevans0)
- headius (Charles Nutter) prefers iteration over search, and recommends Thread.each_backtrace as the method name.
- headius (Charles Nutter) also has an idea for a block form that returns an enumerable, but I think that would greatly increase complexity.
- I don't think each_backtrace is a good name for a method that yields individual caller locations.
- I would like to go forward with Thread.each_caller_location or Kernel.each_caller_location (singleton method only), yielding Thread::Backtrace::Location objects. Is that acceptable?
[Bug #11064] singleton_method for objects with special singleton_class returns an empty array (jeremyevans0)
- It is a bit inconsistent that singleton_method for nil/true/false returns a Method, but singleton_methods does not include the method.
- Is this a bug, or expected behavior?

#7 - 01/04/2022 06:31 AM - mame (Yusuke Endoh)

[Feature #18438] Add Exception#additional_message to show additional error information (mame)
- I'd like to apply error_highlight more exceptions than NameError, but it will break some existing tests that check the return value of Exception#message.
- This feature allows did_you_mean and error_highlight to add their suggestions without modifying the return value of Exception#message.
- To be honest, I'm unsure if this is a right way. I'd like to hear opinions from committers.
- I'll explain some details in the meeting (prefix, needed cooperation, the API style between additional_message and description, etc.)

#8 - 01/05/2022 06:12 PM - zverok (Victor Shepelev)

[Feature #16122] Struct::Value: simple immutable value object (zverok)
- The concept was somewhat favorably discussed a long time ago, but since then no reiterations were made.
- I provided the extended description and rationales and answered the questions that were raised.
- Is it possible to move forward with the idea?

#9 - 01/05/2022 07:25 PM - Dan0042 (Daniel DeLorme)

[Feature #18458] Allow pattern match to set instance variables
- Currently, expr => @var is not supported.
- It doesn't seem like there's a specific reason for that limitation; rescue => @var works fine.
- I think vars should be supported, it would be one less gotcha to learn.
- The same might apply to @@var and $var ?

#10 - 01/07/2022 06:31 AM - mame (Yusuke Endoh)

[Feature #18462] Proposal to merge WASI based WebAssembly support (mame)
- Let's support wasm. This will allow MRI to work on browsers and edge computing platforms.
We were not able to discuss all of the agenda items. We will continue discussion on 28th Jan.

- Status changed from Open to Closed
- Description updated