The next dev meeting

Date: 2020/09/25 13:00-17:00
Place/Sign-up/Agenda/Log: [https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20200925Japan.md](https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20200925Japan.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.

Related issues:

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

History

#1 - 09/01/2020 03:48 PM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 09/02/2020 12:56 AM - marcandre (Marc-Andre Lafortune)

- [Feature #16994] Shorthand syntax for static frozen sets of string/symbols (e.g. %ws{hello world}) (marcandre)
  - Rest of my "Set program", especially: insure interoperability with Array (e.g so array & set works and is efficient)

#3 - 09/03/2020 09:24 AM - Eregon (Benoit Daloze)

- [Bug #17144] Tempfile.open {...} does not unlink the file (eregon)
  - OK to keep the change and make Tempfile.open(&block) finally intuitive? Some usages might need updating but it seems very few.
  - SomeClass.open {...} should always release all resources, Tempfile.open {...} was the only exception.
  - Relying on GC to unlink seems very brittle (even more so on other Ruby implementations).

#4 - 09/03/2020 07:29 PM - jeremyevans0 (Jeremy Evans)
- [Feature #17143] Improve support for warning categories (jeremyevans0)
  - Warning.warn :category keyword support was approved last developer meeting.
  - I would like to add Kernel.warn :category keyword for Ruby-level warnings, and ruby_category_warn[ing] for C-level warnings
  - I have prepared a pull request that adds them, and also adds categories for all core warnings. Is it OK?
- [Bug #16518] Should we rationalize Rational's numerator automatically? (jeremyevans0)
  - I think we should make Kernel#Rational always return a Rational instance, or raise an exception if it cannot.
- [Bug #15712] DateTime=== should be defined and compare date and time instead of just the date (jeremyevans0)
  - DateTime=== is currently the same as Date===, but this seems more accidental than deliberate.
  - Do we want to define DateTime=== and have it be similar to DateTime==?
- [Bug #15661] Disallow concurrent Dir.chdir with block (jeremyevans0)
  - Concurrently, we have a non-verbose warning. Do we want to switch to raising an exception?

#5 - 09/08/2020 03:35 AM - Glass_saga (Masaki Matsushita)

- [Feature #17134] Add resolv_timeout to TCPSocket (glass)
  - It introduces resolve_timeout similar to Socket.tcp. Can I merge it?
  - After merging it, I will work on connect_timeout as well.
- [Feature #15628] init_inetsock_internal should fallback to IPv4 if IPv6 is unreachable (sonalkr132)
  - Should we implement IPv6 to IPv4 fallback or "Happy Eyeballs" (RFC8305) in core?

#6 - 09/08/2020 10:56 AM - fatkodima (Dima Fatko)

- [Feature #14394] Class.descendants (fatkodima)
  - Introduces Module#descendants method as a native way to track Class/Module descendants, instead of inefficient hack like crawling ObjectSpace or tracking descendants via inherited hook. MRI already tracks subclasses, so this won't introduce new overheads.
- [Feature #17056] Array#index: Allow specifying the position to start search as in String#index (fatkodima)
  - If we know from which offset to start searching, this will potentially speed up the index findings.
- [Bug #17038] Enumerable#grep{_v} should be optimized for Regexp (fatkodima)
  - From this issue, it was decided to add a new /f ("fast", the name is debatable) option to regexps.
  - When provided, the methods which previously allocated global MatchData objects, won't do that anymore. This will reduce memory usage overall and greatly speed things up. The benchmark results are provided in the ticket.
- [Feature #15573] Permit zero step in Numeric#step and Range#step (fatkodima)
  - This is for consistency with other behavior. The ticket description describes the problem. Matz responded there that, and I implemented a patch and chose to raise an error on zero step.
- [Feature #13683] Add strict Enumerable#single (fatkodima)
  - This method will ensure the collection contains only one element and return it. This feature is useful and there are many +1 on that.
  - The name for this method (single, one, only, sole) is under discussion
  - Should this method accept a default (like Hash#fetch) when collection is empty, and how: default arg vs block?

#7 - 09/14/2020 06:43 AM - mrkn (Kenta Murata)

- [Feature #14722] python's buffer protocol clone (mrkn)
  - I will fix some bugs until the next meeting.
  - Can I merge this?
- [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:
    
    ```ruby
    [0,1,2,3,4,5] |{(6..0)^-2} | #=> [5, 3, 1]
    # or
    ```
Python follows the former.

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

---

**#8 - 09/18/2020 08:49 AM - byroot (Jean Boussier)**

- [Feature #13381](#) **Expose rb_fstring and its family to C extensions**
  - Still an extremely useful feature long awaited, it would be very disappointing if it didn't make it to 3.0.
  - Both json and msgpack-ruby agreed to a feature that calls String#-@ from C extensions. Meaning rb_fstring_cstr would allow them to saves lots of allocations.

**#9 - 09/18/2020 08:15 PM - Eregon (Benoit Daloze)**

- [Bug #17178](#) **Procs with kw:def/**kw lose elements when called with a single Array (eregon)**
  - Does it sound buggy to you too? Do you think we can experiment and try changing it for Ruby 3? Current behavior seems useless, inconsistent and counter-intuitive.

**#10 - 09/28/2020 11:24 PM - mame (Yusuke Endoh)**

- Status changed from Open to Closed
- Description updated