Ruby master - Misc #15782
DevelopersMeeting20190522Japan
04/22/2019 12:36 PM - k0kubun (Takashi Kokubun)

<table>
<thead>
<tr>
<th>Status</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Please comment your favorite ticket numbers you want to ask to discuss with your SHORT comment or summary. (your summary/comment will help us because we don't need to read all of ticket comments)

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

Date: 2019/05/22 (Wed)
Time: 13:00-17:00 (JST)
Place: Speee Inc. (Tokyo, Japan)
Sign-up: https://ruby.connpass.com/event/129507/
log: https://docs.google.com/document/d/e/2PACX-1vQ4WMPEs7aF9Aqcz5mklUZVF73eQh-VYw0fAE5zLi91nZIIKDKcVSZhPAasn1-eAYKtgHK73qY4yUP5HT/pub
logedit: https://docs.google.com/document/d/1WpcKBnpY6uF23NTJq6uMbbK_3UoxhlMmZc3p4PvNqU/edit#

**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
From non-attendees

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 (ko1)
  * I feel this feature is very useful and some people say +=1: so let discuss about this feature.
Related issues:
Related to Ruby master - Misc #14770: [META] DevelopersMeeting

History

#1 - 04/22/2019 12:37 PM - k0kubun (Takashi Kokubun)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 04/22/2019 12:43 PM - k0kubun (Takashi Kokubun)
- [Misc #14632] git.ruby-lang.org
  - See 5da52d1210. I committed tentative RUBY_REVISION/RUBY_DESCRIPTION behaviors for preview1 as follows. Is it okay?
    - RUBY_REVISION #=> "5da52d1210" (instead of Integer like 67701. 10 chars like email notification and rubyci.)
    - RUBY_DESCRIPTION #=> "ruby 2.7.0dev (2019-04-22 trunk 5da52d1210) [x86_64-linux]" (10 chars, same as the above)

#3 - 04/22/2019 12:48 PM - znz (Kazuhiro NISHIYAMA)
- Description updated
- Subject changed from DevelopersMeeting201905XXJapan to DevelopersMeeting20190522Japan

#4 - 04/23/2019 04:42 AM - tenderlovemaking (Aaron Patterson)
- [Feature #15281] Speed up Set.intersection with size check.
  - Everyone likes faster code. But the question is: are Sets ordered? I think Matz must decide.

#5 - 04/23/2019 03:02 PM - greggzst (Grzegorz Jakubiak)
- [Feature #15323] [PATCH] Proposal: Add Enumerable#filter_map
  - It's expressive as Ruby lang itself
  - Make filter_map(&:predicate) [ |elem| elem * 2 ] syntax possible
  - Should also consider adding filter_map!
  - It enables developers to get rid of such code(below) and has been already accepted by Matz:

```
(1..10).map { |elem| elem * 2 if elem % 2 == 0 }.compact
```

#6 - 04/26/2019 03:52 AM - mame (Yusuke Endoh)
- Description updated

#7 - 04/26/2019 03:52 AM - mame (Yusuke Endoh)
- Description updated

#8 - 04/26/2019 07:11 AM - koic (Koichi ITO)
- [Bug #15745] There is no symmetry in the beginless range and the endless range using Range#inspect (koic)
  - I think that the symmetry of beginless range and endless range becomes more clear by unifying whether it displays nil or not.

#9 - 04/27/2019 03:24 AM - jeremyevans0 (Jeremy Evans)
- [Feature #14915] Deprecate String#crypt
  - I think we should add a real deprecation warning in 2.7 so we can remove it in a later version.

#10 - 04/27/2019 01:26 PM - alanwu (Alan Wu)
- [Feature #15765] [PATCH] Module#name without global constant search
  - This fixes the performance problems for #name on anonymous module / classes, but also comes with some small behavioral changes. Are they okay?

#11 - 04/27/2019 02:23 PM - osyo (manga osyo)
- [Feature #15772] Proposal: Add Time#ceil
  - I want to hear the opinions of developers.
- [Feature #13645] Syntactic sugar for indexing when using the safe navigation operator
  - I presented a use case.
#12 - 04/30/2019 02:54 PM - osyo (manga osyo)

- [Feature #15813] Proposal: Add exception support in Range#first
  - I want to hear the opinions of developers

#13 - 05/01/2019 10:07 AM - methodmissing (Lourens Naudé)

- [Misc #15800] [PATCH] Reduce ONIG_NREGION size from 10 to 4
  - Power of 2 and testing revealed most pattern matches are less than or equal to 4 results
  - MatchData instances over allocate memory for the most common cases
- [Misc #15802] [PATCH] Reduce the minimum string buffer size from 127 to 63 bytes
  - The most common case for string buffers in Rails processes are less than 63 bytes

#14 - 05/01/2019 10:36 PM - ioquatix (Samuel Williams)

- [Feature #14736] Thread selector for flexible cooperative fiber based concurrency
  - Is it something we want to move forward with?
  - Extend hooks for wait_one_pid and Kernel::sleep. Are there other areas to hook into?
  - Add some default implementation using IO.select, e.g. IO::Selector, similar to test spec. Should IO.select accept raw file descriptors?
  - Should IO be non-block by default? Should we adjust behaviour dynamically based on context? How to handle File.read which doesn't give opportunity to assign to nonblock.

#15 - 05/06/2019 08:10 AM - greggzst (Grzegorz Jakubiak)

greggzst (Grzegorz Jakubiak) wrote:

- [Feature #15323] [PATCH] Proposal: Add Enumerable#filter_map
  - It's expressive as Ruby lang itself
  - Make filter_map(&predicate) { [elem] elem * 2 } syntax possible
  - Should also consider adding #filter_map!
  - It enables developers to get rid of such code(below) and has been already accepted by Matz:

```
(1..10).map{ |elem| elem * 2 if elem % 2 == 0 }.compact
```

Updated

#16 - 05/12/2019 12:39 AM - nobu (Nobuyoshi Nakada)

k0kubun (Takashi Kokubun) wrote:

- Can we decide the date to rename the branch "trunk" to "master"? Does anyone have any objection for it? What do we need to prepare for it?
  - Motivation: I've often typo-ed trunk to master even in the git-svn age, because "master" is the standard name for the upstream branch in Git.
  - As yugui (Yuki Sonoda) suggested, should we just use git symbolic-ref refs/remotes/origin/master refs/remotes/origin/trunk?

Just to get rid of a typo, symbolic-ref seems working enough.

#17 - 05/12/2019 08:17 AM - k0kubun (Takashi Kokubun)

(from my previous comment, I changed a plan for the following one and filed another ticket)

- [Misc #15843] Make "trunk" a symbolic-ref of "master" on git.ruby-lang.org
  - Please see "Expected outcome" part. Do you have any objection for it?

#18 - 05/15/2019 09:13 AM - Eregon (Benoit Daloze)

- [Feature #15778] Expose an API to pry-open the stack frames in Ruby
  - Several points were made in favor of Ruby API. Can we make it so this API is clearly marked as "should only be used for debugging"? Are there other concerns?

#19 - 05/15/2019 12:45 PM - hsbt (Hiroshi SHIBATA)

- [Bug #7300] Hash[]  itemCount 1.9.3
  - Can we merge this?

#20 - 05/15/2019 09:46 PM - Eregon (Benoit Daloze)
Future of RubyVM::AST?

What can we do about this? The current situation is confusing for everyone, the API sounds "blessed" by being in core but it's actually experimental and unstable. Can we document it as clearly as possible when this API should be used? Could we improve the API so it's less fragile to parser changes and could be reasonably implemented on other Ruby implementations?

Add Hash#slice! and ENV.slice!(bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Array#extract, Hash#extract, and ENV.extract (bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Hash#slice! and ENV.slice!(bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Array#extract, Hash#extract, and ENV.extract (bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Hash#slice! and ENV.slice!(bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Array#extract, Hash#extract, and ENV.extract (bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Hash#slice! and ENV.slice!(bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Add Array#extract, Hash#extract, and ENV.extract (bogdanvlviv)

I would like to discuss these methods(name, behavior, implementation), and whether we want to add them to Ruby.

Today's venue has been changed to Speee Inc. at Roppongi. Do not go pixiv.

Today's venue has been changed to Speee Inc. at Roppongi. Do not go pixiv.

Today's venue has been changed to Speee Inc. at Roppongi. Do not go pixiv.

Today's venue has been changed to Speee Inc. at Roppongi. Do not go pixiv.