**The next dev meeting**

Date: 2022/03/17 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 2022/03/14. 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 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. 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

---

```ruby
# encoding: utf-8

s = "あいうえおかきくけこ"
s.bytesplice(9, 6, "xx")
p s #=> "あいうxxかきくけこ"
s.bytesplice(2, 3, "x") #=> offset 2 does not land on character boundary (IndexError)
s.bytesplice(3, 4, "x") #=> offset 7 does not land on character boundary (IndexError)
```
#3 - 02/28/2022 09:14 AM - byroot (Jean Boussier)

- [Feature #18576] Rename ASCII-8BIT encoding to BINARY
  - The patch is ready.
  - I'd like to make the change for 3.2.0-preview1, so that we can get early feedback and if it's negative revert before final release.
  - Matz last statement is ambiguous, does it means OK for 3.2.0-preview1?
  - There's an open question on wether C API should alias "ascii8bit" functions too.

#4 - 03/04/2022 05:46 PM - kddeisz (Kevin Newton)

- [Feature #18589] Finer-grained constant invalidation
  - We want to trade a very small memory increase for much better constant cache invalidation.
  - This will allow relying on more stable constant caches for better code specialization in JITs.
  - This can also help speed up initial boot times since caches are currently invalidated each time a class is created.
  - Because of APIs like Object#extend, ActiveRecord::Relation#extending, and autoload, constant caches get busted far more often than they should in production applications, and has a measurable impact on request speed.

#5 - 03/08/2022 01:04 PM - Eregon (Benoit Daloze)

- [Feature #18615] Use -Werror=implicit-function-declaration by default for building C extensions (eregon)
  - OK to do it? It seems an obvious gain and will avoid many confusing errors which happen too late and instead make them clear and early.

#6 - 03/09/2022 07:06 AM - znz (Kazuhiro NISHIYAMA)

- [Feature #18563] Add "graphemes" and "each_grapheme" aliases (znz)
  - Some languages already use graphemes.

#7 - 03/09/2022 07:51 AM - byroot (Jean Boussier)

- [Feature #18566] Merge io-wait gem into core IO class. (byroot)
  - It's very small and any non-trivial IO code will require it.
  - Merge io-wait into io.c for Ruby 3.2
  - Remove io-wait as a dependency of all gems maintained by ruby-core (e.g. net-protocol).
  - Publish a new io-wait version that is simply an empty gem.
  - Add a lib/io/wait.rb stub, with eventually a verbose deprecation warning.
  - It was suggested to do the same with io-nonblock.

#8 - 03/10/2022 09:24 PM - jeremyevans0 (Jeremy Evans)

- [Bug #18620] Not possible to partially curry lambda or proc's call method (jeremyevans0)
  - Should we make changes to allow this to work? It complicates the internals, and the benefits seem minimal.

- [Feature #15357] Proc#parameters returns incomplete type information (jeremyevans0)
  - Last discussed at the March 2020 developer meeting, but no decision was made.
  - I still think it makes sense to consider a lambda keyword argument for Proc#parameters.

- [Feature #12655] Accessing the method visibility (jeremyevans0)
  - This issue discusses many things that have been discussed previously in other issues.
  - I want to focus on one additional method: Module#undefined_minimum_methods
  - We don't currently offer a way to see which methods are undefined in a module.

- [Bug #11063] Special singleton class should return true for singleton_class? test (jeremyevans0)
  - I don't think it is worth changing the existing behavior.
  - Can this be rejected?

#9 - 03/14/2022 05:29 PM - Eregon (Benoit Daloze)

- [Bug #18632] Struct.new wrongly treats a positional Hash as keyword arguments (eregon)
  - Some C functions incorrectly treat a positional Hash as keyword arguments, e.g. Struct.new and Struct#initialize.
  - This is inconsistent with the separation of positional and keyword arguments in Ruby 3
  - Jeremy proposed to audit and fix them

- [Bug #18624] ruby2_keywords does not unmark the hash if the receiving method has a *rest parameter (eregon)
  - This seems a clear inconsistency, and it hurts for 3 reasons:
  - Can cause issues when migrating to other forms of delegation
  - Makes the ruby2_keywords semantics confusing/inconsistent/more complex
  - May force other Ruby implementations to replicate this bug if not fixed
  - I believe we should fix it now. Fixing it later would cause issues when migrating to {*args, **kwargs} or (...).
• [Bug #18633] proc { |a, **kw| a } autosplats and treats empty kwargs specially (eregon)
  ○ Intended or bug? Should we fix it?

#11 - 03/17/2022 02:52 AM - ioquatix (Samuel Williams)
I am in wrong time zone. if you have time, can you check these?

• [Feature #18630] Introduce general IO#timeout and IO#timeout=for all (non-)blocking operations.
  ○ PR is functional. Can we merge it?
• [Bug #18465] Make IO#write atomic.
  ○ When IO is set to sync mode, operations like puts should be atomic.
  ○ Reduce number of times lock is acquired and released per operation.
  ○ Improves consistency of output to sync IO when used by multiple threads.
  ○ PR is functional. Can we merge it?

#12 - 03/22/2022 05:35 AM - mame (Yusuke Endoh)
  - Status changed from Open to Closed
  - Description updated