RubyKaigi 2019 will be at Fukuoka, Apr 18th to 20th 2019. It should be good time to gather Ruby interpreter developers. I keep the meeting space at Fukuoka city, near to the Hakata station (the central station in Fukuoka) before and after RubyKaigi.

- **Date:**
  - 2019/04/17 (Wed), before RubyKaigi2019
  - 2019/04/21 (Sun), after RubyKaigi2019

- **Location**
  - Fukuoka Ruby Content Industry Promotion Center, Fukuoka Higashi Sougouchousha 5th floor, 1 Chome-17-1 Hakataekihigashi, Hakata Ward, Fukuoka

There are two days but Matz can only attend Wednesday.

Maybe on Wednesday we can have a meeting with agenda, for example about Ruby 3.

I'm not sure we need Sunday meetup, but I keep a room for people who want to continue development. If nobody attend Sunday meetup, I'll cancel room reservation.

I hope we can discuss topics on Wed and continue to discuss RubyKaigi days, and make a progress on Sunday.

**Travel support**

Same as last RubyKaigi, Cookpad Inc. will support travel fee (except hotel room fee) so please consider to attend this dev-meeting. Also please invite people who are not a MRI committer but who are important person to talk with us.

I'll announce about this travel support soon.

**Registration**

- Sign-up on [https://rubyassociation.doorkeeper.jp/events/85108](https://rubyassociation.doorkeeper.jp/events/85108)

**Call for Topics**

- Please comment discussion topics on Wed meeting.
- We have limited time to discuss, so that I'll ask Matz which topics we should talk with, and make time schedule in advance if there are many topics.
- Meeting room has a projector so you can bring your slides.
- Maybe attendee's topics will be prioritize.

**Conclusion**

Anyway, please keep your schedule.
FYI: Fukuoka city does not have enough hotel rooms so I recommend you to keep your rooms as soon as possible.

Thanks,
Koichi

**Before**

**log**

- public: [https://docs.google.com/document/d/1hw6Xca8arG6b0V63zvWnNEbIIEEvzS10KXGhzZpl8/pub](https://docs.google.com/document/d/1hw6Xca8arG6b0V63zvWnNEbIIEEvzS10KXGhzZpl8/pub)
- edit: [https://docs.google.com/document/d/1hw6Xca8arG6b0V63zvWnNEIxEvzS10KXGhzZpl8/edit](https://docs.google.com/document/d/1hw6Xca8arG6b0V63zvWnNEIxEvzS10KXGhzZpl8/edit)
Related issues:
Related to Ruby master - Misc #14770: [META] DevelopersMeeting Open
Copied to Ruby master - Misc #16636: DevelopersMeeting before RubyKaigi2020 Closed

History
#1 - 12/27/2018 06:04 AM - ko1 (Koichi Sasada)
ささだです。
日本語でも書いておきます。
RubyKaigi 2019 の前後の日に開発者会議ができればと思っています。前日は午後から、後日はまだ調整中です(まつもとさんは後日はいらっしゃらないと思います)。
・エントリー方法などは、チケットをご覧下さい。
・議論したいトピックなど、お寄せ下さい。
・招待したい方も、ご推薦下さい。
・福岡は宿の手配が大変らしいので、事前の手配をお忘れ無く。
どうぞよろしくお願いします。

#2 - 02/11/2019 12:44 AM - jeremyevans0 (Jeremy Evans)
- [Feature #14183] Keyword Arguments
  - Need to discuss whether to stop converting caller keyword arguments/braceless hash to callee positional argument

I will be able to attend this meeting to discuss this issue.

#3 - 03/18/2019 04:49 AM - ko1 (Koichi Sasada)
"After RubyKaigi hack meeting" is merged with "RubyKaigi 2019 After Hack by Fukuoka.rb!!"
https://fukuokarb.connpass.com/event/124406/

Please sign-up this party if you want to attend it even if you already sign-up on doorkeeper. Details are also written in https://fukuokarb.connpass.com/event/124406/.

Thanks,
Koichi

#4 - 03/18/2019 04:52 AM - ko1 (Koichi Sasada)
Further note:

Thanks,
Koichi

#5 - 03/18/2019 05:12 AM - ko1 (Koichi Sasada)
Schedule on 4/17 (Wed)
- 12:00- door open
- 13:00-18:30 meeting w/Matz.
  - topics is welcome. please comment on this issue.
- 18:30 close

There are several pre-RubyKaigi parties. pls check them.

#6 - 03/24/2019 05:51 AM - Glass_saga (Masaki Matsushita)
[Feature #15553] Addrinfo.getaddrinfo supports timeout

- I would like to have your opinion for patch2.diff

#7 - 04/02/2019 06:34 PM - rafaelfranca (Rafael França)
[Feature #13683] Add strict Enumerable#single

- Matz already showed his discontent with the name single, but only could be a good alternative as proposed by nobu and Lisa on that issue.

#8 - 04/03/2019 12:21 PM - lugray (Lisa Ugray)

05/08/2020
#9 - 04/03/2019 12:24 PM - Eregon (Benoit Daloze)

- [Feature #15752] A dedicated module for experimental features (eregon)
  - We need to address this soon, it prevents clean separation and alternative implementations to implement experimental features.
- [Feature #15743] RubyVM should be renamed to CRuby (eregon)
  - I’d like to gather opinions on this, and ask Matz.

#10 - 04/07/2019 05:21 PM - Eregon (Benoit Daloze)

- [Misc #15723] Reconsider numbered parameters (eregon)
  - We should address this soon. I think it’s good to collect opinions within MRI committers.

#11 - 04/12/2019 01:13 AM - ko1 (Koichi Sasada)

- new built-in class/method definition features in Ruby and C (will be talked at RubyKaigi 2019, my talk) (ko1)

#12 - 04/13/2019 09:43 AM - duerst (Martin Dürst)

- Related to Misc #14770: [META] DevelopersMeeting added

#13 - 04/13/2019 09:46 AM - duerst (Martin Dürst)

- [Bug #15764] Whitespace and control characters should not be permitted in tokens
  (This can and should be easily fixed, but I’d like Matz to give it a check.)

#14 - 04/13/2019 02:10 PM - ktsj (Kazuki Tsujimoto)

- [Feature #14912] Introduce pattern matching syntax
  - I’d like to introduce latest specification, discuss about the details.

#15 - 04/14/2019 01:12 AM - ko1 (Koichi Sasada)

Hi all,

I asked Matz and we fixed dev-meeting schedule:

- 12:00-13:20 door open. I (ko1) will be there with bento-lunch.
- 13:20-13:30 opening
- 13:30-14:00 (1) keyword arguments progress report (mame)
- 14:00-14:30 (2) static checking progress report (mame)
- 14:30-15:00 (3) pattern matching (k_tsj)
- 15:00-15:30 break
- 15:30-16:00 (4) concurrency (matz)
- 16:00-16:30 (5) new built-in class/method definition (ko1)
- 16:30-17:00 break
- 17:00-18:00 (6) RubyVM + numbered parameters (eregon)
- 18:00- other remaining topics

For speakers: please prepare your slide at dev meeting. maybe 15min talk is good to reserve discussion time.

As you can see, each time slot is 30 min, so it is hard to discuss enough. So this dev-meeting is sharing ideas with committers and discuss more in break time and RubyKaigi dates.

Time schedule will not be managed strictly, but it is easy to exceeds. I ask you cooperate us to finish meeting successfully.

Thanks,
Koichi

#16 - 04/15/2019 12:01 AM - ko1 (Koichi Sasada)

12:00-13:20 door open. I (ko1) will be there with bento-lunch.

ko1 will be there with my own bento (to welcome you).
If you want to eat with me, please bring your own bento (lunch).

I heard that Fukuoka has very nice food restaurant. Enjoy.
Thanks,
Koichi

#17 - 04/16/2019 08:44 AM - ko1 (Koichi Sasada)

12:00-13:20 door open.

My flight will arrive Fukuoka 11:35. So it seems difficult to open at 12:00. It depends on real flight schedule.
If I'm not available please take a sheet in a place (there are some chairs at 5F, Fukuoka Ruby Content Industry Promotion Center).

- Fukuoka Ruby Content Industry Promotion Center, Fukuoka Higashi Sougouchousha 5th floor, 1 Chome-17-1 Hakataekihigashi, Hakata Ward, Fukuoka
- 福岡県Ruby・コンテンツ産業振興センター 5階 (in Japanese)

http://frac.jp/about/ (Japanese page)

Thanks,
Koichi

#18 - 04/16/2019 03:07 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.

#19 - 04/16/2019 05:24 PM - ko1 (Koichi Sasada)

16:00-16:30 (5) new built-in class/method definition (ko1)

Sorry I couldn't make a new ticket about it. I write my idea here.

Japanese article: https://techlife.cookpad.com/entry/2019/04/17/014142

The basic idea is writing built-in methods in Ruby with FFI.

class String
  def length
    __C__.str_length
  end

  def replace other
    __C__.str_replace other
  end
end

In this case, str_length() C function is called and str_replace function is called with a parameter other. You don't need to write rb_define_method() in C code.

There is several advantages:

- You can write complex features like kwparm, exception and so on (in Ruby)
- You can add DSL like attribute, like that:

class String
  def length
    __ATTR__.pure
    __C__.str_length
  end
end

In this case, we can annotate String#length has no side-effect with Ruby DSL.

This idea can introduce additional overhead compared with traditional C method definitions. On my Ruby2019 talk, two technical achievements are introduced and I think this approach is feasible from performance perspective.

#20 - 04/17/2019 01:03 AM - alanwu (Alan Wu)

05/08/2020
Looking for feedback

[Bug #15645] It is possible to escape Mutex#synchronize without releasing the mutex

- The suggestion posted as a comment seems to make sense. Could someone take a look and possibly merge it in?

#21 - 04/17/2019 04:37 AM - naruse (Yui NARUSE)
- Description updated
- Subject changed from Developers Meeting before/after RubyKaigi2019 to DevelopersMeeting before/after RubyKaigi2019

#22 - 04/22/2019 12:34 PM - k0kubun (Takashi Kokubun)
- Status changed from Open to Closed

#23 - 02/17/2020 08:28 AM - ko1 (Koichi Sasada)
- Copied to Misc #16636: DevelopersMeeting before RubyKaigi2020 added