

Ruby 2.6 Update

Kazuhiro NISHIYAMA

第85回 Ruby関西 勉強会

2019/02/16

株式会社Ruby開発

自己紹介

- 西山 和広
- Ruby のコミッター
- twitter, github など: @znz
- 株式会社Ruby開発 www.ruby-dev.jp

ITアイランドセミナー in 姫島2019

- 2019-03-15(金)～16(土) 大分県姫島
- <https://connpass.com/event/115052/>
- <https://www.ruby-dev.jp/seminar/himeshima/2019>

OSS Gate

- OSS Gate大阪ワークショップ2019-03-09
<https://oss-gate.doorkeeper.jp/events/86154>
- 「OSSの開発に参加する」を実際に体験するワークショップ
- 実際にOSSの開発に参加する人（「ビギナー」）と「ビギナー」をサポートする人（「サポーター」）を募集中

agenda

- 2.6.0 での変更の紹介
- 2.6.0 での問題
- 2.6.1 での問題
- 2.6.2, 2.6.3 の予定

オススメ記事

- プロと読み解く Ruby 2.6 NEWS ファイル
- クックパッド開発者ブログ
<https://techlife.cookpad.com/entry/2018/12/25/110240>

2.6.0 での 変更の紹介

\$SAFE がプロセスグローバルに

- Proc#call で保存されなくなった
- スレッドローカル (Fiber ローカル) ではない
 - マルチスレッドプログラムでスレッドセーフに扱えなくなった
- \$SAFE = 1 から \$SAFE = 0 で戻せるようになった

終端なしの Range

```
ary[1..] #=> ary[1..-1] と同じ
```

```
(1..).each {|index| ... } # 1 から無限にループ
```

```
# each_with_index を 0 でなく 1 から始める  
ary.zip(1..) {|elem, index| ... }
```

キーワード引数とオプション 引数のコーナーケースを禁止

```
def foo(h = {}, key: :default)
  p [h, key]
end
```

```
foo(:key => 1, "str" => 2)
#=> [{"str"=>2}, 1] (2.5 まで)
#=> non-symbol key in keyword arguments:
#   "str" (ArgumentError) (2.6)
```

ローカル変数の shadowing 警告を削除

以下のような例で shadowing outer local variable 警告がでなくなった

```
user = users.find { |user| cond(user) }
```

バックトレース表示

プロセス終了時のバックトレースで cause のバックトレースも表示されるようになった

```
$ ruby -e 'def a;b;rescue;raise "in a";end;def b;raise "in b";end;a'
Traceback (most recent call last):
  2: from -e:1:in `<main>'
  1: from -e:1:in `a'
-e:1:in `b': in b (RuntimeError)
  2: from -e:1:in `<main>'
  1: from -e:1:in `a'
-e:1:in `rescue in a': in a (RuntimeError)
```

flip-flop が deprecated

条件式としての範囲式

https://docs.ruby-lang.org/ja/2.6.0/doc/spec=2foperator.html#range_cond

```
5.times{|n|
  if (n==2)..(n==3)
    p n
  end
}
#=> 2
# 3
```

to_h がブロックを受け取るように

従来の to_h の使い方

```
["Foo", "Bar"].map {|x| [x.upcase, x.downcase] }.to_h  
#=> {"FOO"=>"foo", "BAR"=>"bar"}
```

新しい用法

```
["Foo", "Bar"].to_h {|x| [x.upcase, x.downcase] }  
#=> {"FOO"=>"foo", "BAR"=>"bar"}
```

Array#filter, Array#filter!, Enumerable#filter

- Array#select, Array#select!,
Enumerable#select の別名
- (ruby 1.6 までの Array#filter は今の
Array#map! と同じ)
- (ruby 1.8 から 2.5 には Array#filter はない
)

Dir#each_child, Dir#children

.,.. を含まない Dir.each_child,
Dir.children のインスタンスメソッド版

Enumerable#chain

```
a1 = %w(1 2)
a2 = %w(3 4)
a3 = %w(5 6)
[a1, a2, a3].each{|ary| ary.each{|e| p e}} # 多重ループ
(a1+a2+a3).each{|e| p e} # 配列の時のみ
a1.chain(a2, a3).each{|e| p e}
(a1.each + a2.each + a3.each).each{|e| p e}
a1.chain(a2.reverse_each, a3).each{|e| p e} # each を持てば繋がられる
```

Enumerator::ArithmeticSequence

- 等差数列を提供するためのクラス
- Python のスライスに相当
- いろんなライブラリが対応していけばより便利に

```
3.step(by: 2, to: 10)  
(3..10)%2
```

Exception#full_message の引数追加

- :highlight, :order
- ruby 2.5.1 にもバックポート済み

open のモードに x 追加

File::EXCL の代わりに使える x 追加 (C11 由来)

Kernel#then

Kernel#yield_self の別名

Kernel#Integer など に :exception

```
Integer('hello')  
#=> `Integer': invalid value for Integer(): "hello" (ArgumentError)  
Integer('hello', exception: false) #=> nil  
p system("ruby -e raise") #=> false  
p system("ruby -e raise", exception: true)  
#=> `system': Command failed with exit 1: ruby -e raise (RuntimeError)
```

system, exec などの close_others

- デフォルトが false に
- ruby 本体が開く fd は以前と変わらず
FD_CLOEXEC を設定
- 拡張ライブラリが開く fd やプロセス起動時
に継承した fd に影響

KeyError.new, NameError.new

- :receiver, :key 追加
- Ruby レベルでも設定可能に
- did_you_mean との相性が良くなる

関数合成オペレータ

Proc#<<, Proc#>>

```
plus2  = -> x { x + 2 }  
times3 = -> x { x * 3 }
```

```
times3plus2 = plus2 << times3  
p times3plus2.(3) #=> 3 * 3 + 2 => 11  
p times3plus2.(4) #=> 4 * 3 + 2 => 14
```

```
plus2times3 = times3 << plus2  
p plus2times3.(3) #=> (3 + 2) * 3 => 15  
p plus2times3.(5) #=> (5 + 2) * 3 => 21
```

String#split がブロック対応

```
"foo/bar/baz".split("/") {|s| p s } #=> "foo", "bar", "baz"
```

2.6.0 で

の問題

2.6.0 の Net::HTTP のバグ

- 詳細: <https://mensfeld.pl/2019/01/exploring-a-critical-netprotocol-issue-in-ruby-2-6-0p0-and-how-it-can-lead-to-a-security-problem/>

2.6.0 の Net::HTTP のバグ

- <https://bugs.ruby-lang.org/issues/15468>
- マルチバイト文字の扱いの問題
- String#byteslice を使うべきところで String#[] を使っていた
- 使い方によってはセキュリティ問題が起きる
- 2.6.1 では修正済み

2.6.1 で

の問題

2.6 で bundler が default gem に

default gem とは？

- default gem : 標準添付ライブラリーだが
gem で新しいバージョンに更新可能
- bundled gem : ruby と一緒にインストール
される gem (アンインストールも可能)

bundler の不具合

- Ruby 2.6.1 の Bundler の不具合のお知らせ

[https://www.hsbt.org/
diary/20190205.html#p02](https://www.hsbt.org/diary/20190205.html#p02)

bundler の不具合

● <https://bugs.ruby-lang.org/issues/15582>

● 1.17.2 そのままなら問題ない

● 2.0.1 や 1.7.3 をインストールすると壊れる

● <https://bugs.ruby-lang.org/issues/15469>

● default gem (csv, json など) の別バージョンが壊れない

bundler: 対処方法

- 2.6.2 を待つ
- `gem update --system`
- `rbenv + ruby-build` なら以下のようにパッチを適用しつつインストール

```
curl -sSL \  
https://bugs.ruby-lang.org/attachments/download/7631/15582-bundler-gemspec.patch \  
https://bugs.ruby-lang.org/attachments/download/7635/r15469-bundler-final.patch |  
rbenv install --patch 2.6.1
```

2.6.2, 2.6.3

の予定

今後のリリース予定

- 2.6.2 で Unicode 12.0 対応予定 (3月?)
- 2.6.3 で Unicode 12.1 対応 (新元号対応) 予定 (4月?)
- (別リリースが挟まってバージョンがずれる可能性はある)

もっと先の話

- 2.3 から入った
frozen_string_literal は 3.0 ではデフォルトにならない
- キーワード引数が分離されるはず (*rest と **kwrest が混ざらなくなる)
- パターンマッチが入るかも
<https://bugs.ruby-lang.org/issues/14912>