

Ruby 考古学 消された機能編

Archaeology of Ruby:
Removed Features

Kazuhiro NISHIYAMA

*RubyKaigi2014
2014/09/19*

Self introduction

- one of Ruby Committers
 - Mr. fix typo (by Chikanaga)
- I use Ruby since 1.4

Agenda

- **リリースに入った後、消された機能**

Some features removed after release

- **開発版のみに入っていた機能**

Some features only in development branch

Safe Level

- \$SAFE=4 is obsolete since 2.1.0
- 不完全な Sand Box
- Safe Level 自体の削除の話も何度かあった

Safe level

- [Feature #8468 Remove \\$SAFE](#)
 - Matz agreed at GitHub Tokyo drinkup (on 2013)
 - tDiary, which is the only application using `$SAFE == 4`, also agreed
 - `$SAFE < 4` is a fail-safe feature to detect bugs of applications

Safe Level

- `$SAFE=4` is obsolete since 2.1.0
raise `ArgumentError`
- `untrust` since 1.9.1
- `untrust` is deprecated since 2.1.0
its behavior is same as `taint`

' .' from \$LOAD_PATH

- security reason
- same as PATH does not contain ' .' since 1.9.2
- use `require_relative` instead

\$KCODE

- from \$KCODE to Encoding (m17n)
- \$KCODE #=> nil # warning: variable \$KCODE is no longer effective
- \$KCODE= # warning: variable \$KCODE is no longer effective; ignored

Regex options

- Regex literal options
(remains) : /n /e /s /u

```
//n.encoding #=> #<Encoding:US-ASCII>  
//e.encoding #=> #<Encoding:EUC-JP>  
//s.encoding #=> #<Encoding:Windows-31J>  
//u.encoding #=> #<Encoding:UTF-8>
```

`$=` (ignore case)

- `$= = true`
- ruby 1.8.7: warning: modifying `$=` is deprecated
- ruby 1.9.3: warning: variable `$=` is no longer effective; ignored

\$deferr

between 1.6.7 and 1.8.0

- `$deferr` → `$stderr`
- `$defout` → `$stdout`
- `$stderr=` → `STDERR.reopen`
- `$stdin=` → `STDIN.reopen`
- `$stdout=` → `STDOUT.reopen`

if cond:

until 1.8

```
if true:  
  p true  
end
```

: をつけられる隠し機能の削除

File.exists?

- File.exist? (recommend)
- File.exists? (deprecated)

Ruby 本体のメソッド名は出来るだけ三単現のsを付けない

```
File.exists? IO::NULL
# warning: File.exists? is a deprecated
# name, use File.exist? instead
```

Agenda

- リリースに入った後、消された機能

Some features removed after release

- **開発版のみに入っていた機能**

Some features only in development branch

Symbol < String in 2006 (1.9.0-dev)

- Symbol が String を継承していた時期があった
- Symbol < String も止める。
caseとかでのバグをたくさん生んでしまう

<http://www.rubyist.net/~matz/20061107.html#p03>

Symbol < String (cont.)

```
ruby 1.8.7:  
  // =~ :s  
  #~> TypeError: can't convert Symbol into String  
  methods[0].class #=> String
```

```
ruby 1.9.3:  
  // =~ :s #=> 0  
  methods[0].class #=> Symbol
```

- 継承は止めても String に似た扱いは入った。

__send

between 1.8 and 1.9

- `__send__` cannot call private methods
- `__send`, `__send!`
- `public_send` since 1.9.1

private メソッド呼び出し用途が多く影響範囲が広いいため元の挙動に

Real multi-value (to_splat)

1.9.x

- `to_splat` (`!= to_ary`) (`!= to_a`)
- `svalue`, `avalue`, `mvalue`
- e.g. `a, b = mvalue`

Symbol.find in 2.2.0-dev

- Symbol.find(str)
 - > symbol or nil
 - Return the related symbol if the symbol already exists.
 - Return nil if not.
- revision 47543
 - Removed because of Symbol GC
 - If you still want this, request again on Redmine.

statfs in 2.2.0-dev

- IO#statfs and File::Statfs
 - <https://bugs.ruby-lang.org/issues/9772>
 - statfs.f_type for tests
- Reject at [DevelopersMeeting20140517 Japan](#)

「Matz: 色々込み入ってるので core には入れないで test 配下へ. 欲しいということがあったら gem にしてください。」

Conclusion

- Some features only in development branch
- Some features cannot introduce because of compatibility