

あまり知られていないRuby の便利機能

Kazuhiro NISHIYAMA

鹿児島Ruby会議01

2019/11/30

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

String#undump

- String#dump \Leftrightarrow String#undump
 - since ruby 2.5
- String#dump \neq String#inspect
- String#undump \neq eval

Hash#transform_*

- convert from Hash to Hash
- Hash#transform_values{|v|...}
 - since ruby 2.4
- Hash#transform_keys{|k|...}
 - since ruby 2.5

Hash#to_h with block

- Hash#to_h{|k, v| ... }
- with block since ruby 2.6
- without block since ruby 2.0.0

warn with uplevel:

- old: warn "#{caller(1, 1)[0]}: warning: message"
- new: warn "message", uplevel: 1
 - since ruby 2.5

abort(message)

- abort("failed message")
- \equiv warn("failed message");
exit(false)

rand(range)

- rand(range)

 - rand(1..6)

 - since ruby 1.9.3

- NG: rand(endless_range)

 - rand(1..)

 - Errno::EDOM (Numerical argument out of domain)

String.new

- `String.new.encoding` → ASCII-8BIT
- `String.new(encoding: 'euc-jp').encoding` → EUC-JP
- `'' .dup` → UTF-8 (script encoding)
- `+' '` → UTF-8 (script encoding)
 - `'' .+@` (for method chain)
- (useful with frozen string literal)

String#gsub(pattern, hash)

```
string.gsub(/['&"<>]/, {  
  "'" => '&#39;',  
  '&' => '&amp;',  
  '"' => '&quot;',  
  '<' => '&lt;',  
  '>' => '&gt;',  
})
```

Regex.union

```
Regex.union                #=> /(?!)/
Regex.union("penzance")   #=> /penzance/
Regex.union("a+b*c")      #=> /a\+b\*c/
Regex.union("skiing", "sledding")
Regex.union(["skiing", "sledding"])
  #=> /skiing|sledding/
Regex.union(/dogs/, /cats/i)
  #=> /(?-mix:dogs)|(?!-mx:cats)/
```

String#*_with?

```
"hello".start_with?("hell")           #=> true
"hello".start_with?(/H/i)              #=> true

# returns true if one of the prefixes matches.
"hello".start_with?("heaven", "hell")  #=> true
"hello".start_with?("heaven", "paradise") #=> false

"hello".end_with?("ello")              #=> true

# returns true if one of the +suffixes+ matches.
"hello".end_with?("heaven", "ello")    #=> true
"hello".end_with?("heaven", "paradise") #=> false
```

🔴 NG: starts_with?, ends_with?

String# { prepend, delete_prefix, delete_suffix, chomp, chop }

```
"end".prepend("prep")      #=> "prepend"  
"prefix".delete_prefix("pre") #=> "fix"  
"suffix".delete_suffix("fix") #=> "suf"  
"suffix".chomp("fix")     #=> "fix"  
"hello\r\n".chomp        #=> "hello"  
"hello\r\n".chop         #=> "hello"
```

String# {delete, tr}

```
"hello".delete "l","lo"           #=> "heo"  
"hello".delete "lo"              #=> "he"  
"hello".delete "aeiou", "^e"     #=> "hell"  
"hello".delete "ej-m"           #=> "ho"  
"hello".tr('el', 'ip')          #=> "hippo"  
"hello".tr('a-y', 'b-z')       #=> "ifmmp"  
"hello".tr('^aeiou', '*')       #=> "*e**o"
```

参考文献

- リファレンスマニュアル
<https://docs.ruby-lang.org/ja/>
- 間違いなどを見つけたら
<https://github.com/rurema/doctree>
- もっと気軽に確認したいなら
<https://ruby-jp.github.io/>
ruby-jp Slack の #rurema など