2019年12月7日

Ruby初級者向けレッスン 72回
― 文字列 ―

ひがき @ Ruby関西

文字列

- リテラル
- 文字列操作
- エンコーディング
- 順序
- 比較
- 数え上げ
- 破壊

文字列リテラル

```
"Ruby関西"                    # => "Ruby関西"
"Ruby関西".class              # => String

'文字列に " を含む'
                              # => "文字列に \" を含む"

%|' も " も含む|
                              # => "' も \" も含む"
```

式展開

```
name = '松本行弘'
age  = 54

"#{name}さん(#{age})"
                    # => "松本行弘さん(54)"

'#{name}さん(#{age})'
              # => "\#{name}さん(\#{age})"
```

文字列操作

```
"Ruby" + "関西"        # => "Ruby関西"

"こんにちは" * 2
              # => "こんにちはこんにちは"

"hello world".sub(/ello/, 'ard')
                      # => "hard world"
```

エンコーディング

```
s = "Ruby関西"
s.encoding       # => #<Encoding:UTF-8>

e = s.encode(Encoding::EUC_JP)
e.encoding       # => #<Encoding:EUC-JP>

Encoding.list
```

マジックコメント

- 指定がなければ `UTF_8`
- **Emacs**

```
# -*- coding: cp932; -*-
```

- **Vim**

```
# vi: set fileencoding=cp932 :
```

順序

```
["Ruby", "Java", "C#", "Lisp"].sort
    # => ["C#", "Java", "Lisp", "Ruby"]

"a".succ                # => "b"
"b".succ.succ           # => "d"
"z".succ                # => "aa"
"9".succ                # => "10"
"Ruby".succ             # => "Rubz"
```

比較

```
"Ruby" == "Ruby"          # => true
"Ruby" != "Ruby"          # => false
"Ruby" <=> "Ruby"         # => 0
"Ruby" === "Ruby"         # => true
"Ruby".eql? "Ruby"        # => true
"Ruby".equal? "Ruby"      # => false
"Ruby" =~ /Ruby/          # => 0
"Ruby" !~ /Ruby/          # => false
```

比較 <=>

順序判定

```
"b" <=> "a"          # =>  1
"b" <=> "b"          # =>  0
"b" <=> "c"          # => -1
```

比較 ===

```
case str
when "foo"
    ...
end

"foo" === str
```

比較 eql?

```
h = {"foo" => "value"}
h["foo"]                    # => "value"
h["bar"]

# "foo".hash == "bar".hash ならば ...

"bar".eql? "foo"            # => false
h["bar"]                    # => nil
```

比較 equal?

```
s = "Ruby"
t = "Ruby"

s == t                  # => true
s.equal? t              # => false

s.object_id             # => 70273420555100
t.object_id             # => 70273420555080
```

比較 正規表現

```
"Ruby" =~ /Ruby/          # => 0
"Ruby" =~ /u/             # => 1
"Ruby" =~ /b/             # => 2

"Ruby" =~ /Python/        # => nil

/Ruby/ =~ "Ruby"          # => 0
/Ruby/ =~ "Python"        # => nil
```

比較　エンコーディング

```
utf8 = "Ruby関西"
utf8.encoding      # => #<Encoding:UTF-8>

sjis = utf8.encode(Encoding::CP932)
euc  = utf8.encode(Encoding::EUC_JP)

utf8 == sjis                    # => false
sjis == euc                     # => false
utf8 == euc                     # => false
```

数え上げ

```
"Ruby"[0]                  # => "R"
"Ruby"[1]                  # => "u"
"Ruby"[2]                  # => "b"
"Ruby"[-1]                 # => "y"

"Ruby関西"[1, 2]           # => "ub"
"Ruby関西"[2..3]           # => "by"
"Ruby関西"[3..-2]          # => "y関"
```

数え上げ (2)

```
puts "Ruby関西".chars


# >> R
# >> u
# >> b
# >> y
# >> 関
# >> 西
```

数え上げ (3)

```
"Ruby関西".bytes
     # => [ 82, 117,  98, 121, 233,
            150, 162, 232, 165, 191]

"R".ord                  # => 82
 82.chr                  # => "R"
```

数え上げ (4)

```
s = <<EOF
No
Ruby
No
Life
EOF

s          # => "No\nRuby\nNo\nLife\n"
s.lines  # => ["No\n", "Ruby\n", "No\n", "Life\n"]
```

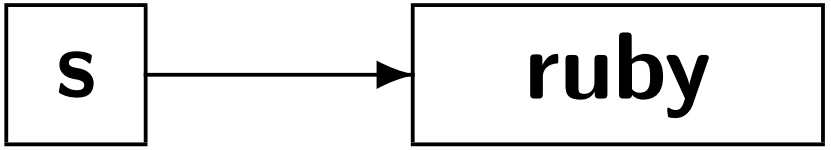
破壊

```
s = "ruby"
s.upcase                    # => "RUBY"
s                           # => "ruby"

s.upcase!                   # => "RUBY"
s                           # => "RUBY"
```

破壊 (2)

```
s = "ruby"
```

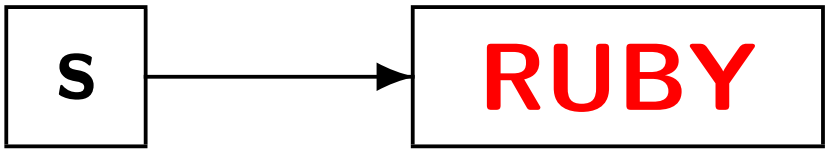



```
s.upcase
```

RUBY

破壊 (3)

```
s = "ruby"
```




```
s.upcase!
```

破壊 (4)

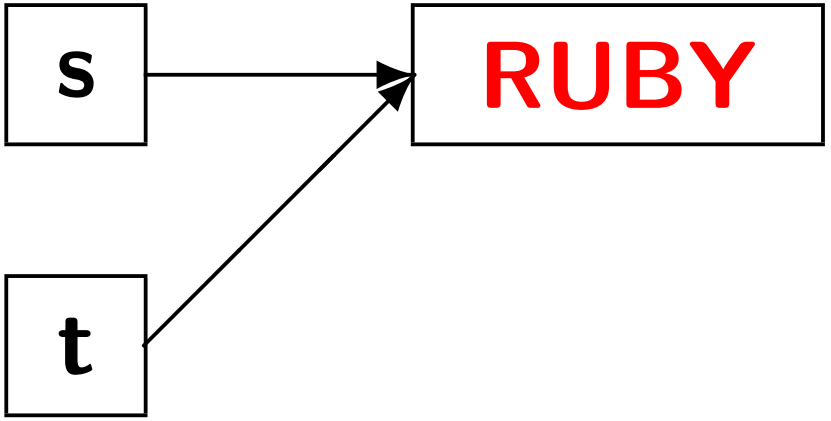
```
s = "ruby"
t = s                   # => "ruby"
t.upcase!               # => "RUBY"
s                       # => "RUBY"
```

破壊 (5)

```
s = "ruby"

t = s

t.upcase!
```



破壊を防ぐには

```
s = "ruby"
s.freeze


s.upcase!
# ~> -:4:in `upcase!': can't modify
    frozen String (FrozenError)
```

破壊を防ぐには (2)

マジックコメント

```
# frozen_string_literal: true
s = "ruby"

s.upcase!
# ~> -:4:in ‘upcase!’: can’t modify
    frozen String (FrozenError)
```

演習問題 0

今日のレッスンで分からなかったこと、疑問に思ったことをグループで話し合ってみよう。

演習問題 1

文字列の

1. 行数
2. 単語数
3. 文字数
4. バイト数

を数えてみよう。

演習問題 2

文字列の

1. 単語の出現回数
2. 文字の出現回数

を数えてみよう。

演習問題 3

こにんちは みさなん おんげき ですか？ わしたは げんき です 。この ぶしんょう は イリギス の ケブンッリジ だがいく の けきんゅう の けっか にげんん は もじ を にしんき する とき その さしいょ と さいご の もさじえ あてっれいば じんゅばん は めゃちちくゃ でも ちんゃと よめる と いう けきんゅう に もづといて わざと もじの じんゅばん を いかれえて あまりす 。 どでうす？ ちんゃと よちめゃう でしょ？ ちんゃと よためら はのんう よしろく

自己紹介

- 名前 (ニックネーム)
- 普段の仕事・研究内容・代表作
- Ruby歴・コンピュータ歴
- 勉強会に来た目的
- などなど

参考

- 解答例

  https://github.com/higaki/learn_ruby_kansai_88

- るりま

  https://docs.ruby-lang.org/ja/