Ruby 初級者向けレッスン in KOF 2005

かずひこ@株式会社 ネットワーク応用通信研究所

2005 年 10 月 29 日

Ruby のイテレータを学ぼう

- イテレータを使おう
- イテレータを作ろう

イテレータとは？

- 「繰り返し」を行うためのメソッド
- Ruby の特徴的な要素の一つ

配列とイテレータ

- 配列の要素を順に処理する each メソッド

```
# each_sample.rb
["dog", "cat", "cow"].each do |elem|
  p(elem)
end

$ ruby each_sample.rb
"dog"
"cat"
"cow"
```

- [”dog”, ”cat”, ”cow”] という配列オブジェクトの each メソッドが「do |elem| p(elem); end」というブロックを引数として呼ばれています
- |〜| で囲まれた部分をブロックパラメータと言います
- p はオブジェクトを表示するメソッドです

ハッシュとイテレータ

- ハッシュにも each が使えます

```
# each_sample2.rb
{"dog" => 1, "cat" => 2, "cow" => 3}.each do |elem|
  p(elem)
end

$ ruby each_sample2.rb
["cow", 3]
["cat", 2]
["dog", 1]
```

- ハッシュの key を取り出すには？

```
# each_sample2.rb
{"dog" => 1, "cat" => 2, "cow" => 3}.each do |elem|
  p(elem[0])
end
```

- ↑ちょっとわかりにくい...

```
# each_sample2.rb
{"dog" => 1, "cat" => 2, "cow" => 3}.each do |key, value|
  p(key)
end

$ ruby each_sample2.rb
"cow"
"cat"
"dog"
```

- ブロックパラメータは複数持つことができます

```
# each_sample2.rb
{"dog" => 1, "cat" => 2, "cow" => 3}.each_key do |key|
  p(key)
end
```

- Hash クラスには each_key とか each_value というメソッドもあります

Enumerable モジュール

- each を用いてさまざまなイテレータが定義されています
- each が定義されているクラスにインクルードすれば、各種イテレータが使えます
- Array や Hash には最初からインクルードされています

Enumerable モジュールで使えるイテレータ

- collect (または map)
  - 各要素に対してブロックを評価した結果を全て含む配列を返します

```
ary = [1, 2, 3, 4, 5].collect do |elem|
  elem * 2
end
p(ary)

[2, 4, 6, 8, 10] # 各要素を 2 倍した配列
```

- select (または find_all)
  - 各要素に対してブロックを評価した値が真であった要素を全て含む配列を返します

```
ary = [1, 2, 3, 4, 5].select do |elem|
  elem % 2 == 0
end
p(ary)

[2, 4] # 偶数だけ抜き出した配列
```

- reject
  - 各要素に対してブロックを評価し、その値が偽であった要素を集めた新しい配列を返します
  - select の反対

```
ary = [1, 2, 3, 4, 5].reject do |elem|
  elem % 2 == 0
end
p(ary)

[1, 3, 5] # 偶数以外を抜き出した配列
```

- detect(または find)

  - 要素に対してブロックを評価した値が真になった最初の要素を返します

```
ans = [1, 2, 3, 4, 5].detect do |elem|
  elem % 2 == 0
end
p(ans)

2 # 最初の偶数
```

イテレータを作ろう

- ブロックを受け取って、各要素に対してブロックを呼び出します
- ブロックを呼び出すには yield 式を使います

yield の動作

- メソッド内で yield を呼ぶと、そのたびにメソッドに指定したブロックが起動されます

```
# yield_test.rb
def three_times
  yield
  yield
  yield
end

three_times do
  p("hello!")
end

$ ruby yield-test.rb
"hello!"
"hello!"
"hello!"
```

- yield の引数がブロックパラメータとなります

```
# yield_test.rb
def one_two_three
  yield(1)
  yield(2)
  yield(3)
end
```

```
one_two_three do |arg|
  p(arg)
end

$ ruby yield-test.rb
1
2
3
```

イテレータの例

- collect **メソッド**
  - **各要素に対してブロックを評価した結果を全て含む配列を返します**

```
class Array
  def collect
    ret = []
    self.each do |elem|
      ret << yield(elem)
    end
    return ret
  end
end
```

演習問題

- **以下のイテレータメソッドを、自分で定義してみましょう**
  - select
  - reject
  - detect

参考文献

- **『たのしい** Ruby』 ISBN:4797314087
- **『プログラミング** Ruby』 ISBN:4894714531

今後の情報源

公式 Web サイト
: http://www.ruby-lang.org/

リファレンスマニュアル
: http://www.ruby-lang.org/ja/man/

日本 Ruby の会
: http://jp.rubyist.net/

Rubyist Magazine
: http://jp.rubyist.net/magazine/

ふぇみにん日記
: http://kazuhiko.tdiary.net/