

## Ruby 初級者向けレッスン第7回

かずひこ@株式会社 ネットワーク応用通信研究所, コウザイ@小波ゼミ

2006 年 5 月 13 日

### 今日のメニュー

- Rubyist の基礎知識
- イテレータの使い方
- イテレータの使用例

### Rubyist の基礎知識

- 調べ方
- 実行の仕方

### 調べ方

- ウェブのリファレンスマニュアル
  - <http://www.ruby-lang.org/ja/man/>
- ri ( 英語 )
  - Ruby に同梱されている
- refe ( 日本語 )
  - <http://i.loveruby.net/ja/prog/refe.html>
- rbbr ( 日本語・英語 )
  - <http://ruby-gnome2.sourceforge.jp/hiki.cgi?rbbr>

### ri の使い方

- クラスの説明
  - ( 例 ) ri Array
- メソッドの説明
  - ( 例 ) ri Array.new
  - ( 例 ) ri Array#each
- あいまいに探す
  - ( 例 ) ri A
  - ( 例 ) ri A.e
- ヘルプの表示
  - ri -help
- クラス・モジュールの一覧
  - ri -c
- クラス・モジュール・メソッドの一覧
  - ri -l

### Ruby プログラムの実行の仕方

- ruby ファイル名
- irb

### irb

- 対話型の Ruby
- 式を入力すると、すぐに実行して式の値を表示します

```
$ irb
irb(main):001:0> 1 + 2    式を入力
=> 3    式の値を出力
irb(main):002:0> puts "Hello"    式を入力
Hello    puts の実行結果
=> nil    式の値を出力 ( puts の返り値は nil )
```

## イテレータの使い方

- イテレータとは？
- 配列とイテレータ
- ハッシュとイテレータ
- Enumerable モジュール

## 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

## 単純な繰り返し

- ある回数繰り返す

```
# times_sample.rb
3.times do
  puts("Hello.")
end

$ ruby times_sample.rb
Hello.
Hello.
Hello.
```

- 3 という整数オブジェクトの times メソッドが「do puts("Hello.") end」というブロックを引数として呼ばれています
- do ~ end で囲まれた部分をブロックと言います

## 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

## 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]
```

## ハッシュとイテレータ

- 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

## ブロックパラメータと変数

- ブロックパラメータと同じ名前の変数が前にあったらそれを使います

```
# param_sample1.rb
i = 10
[1, 2, 3].each do |i| end
p i
```

```
$ ruby param_sample1.rb
3
```

- ブロックパラメータと同じ名前の変数が前になければブロックの外では使えません

```
# param_sample2.rb
[1, 2, 3].each do |i| end
p i
```

```
$ ruby param_sample2.rb
param_sample2.rb:2: undefined local variable or method 'i' for main:Object (NameError)
```

## 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

## Enumerable モジュールで使えるイテレータ

- 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

## Enumerable モジュールで使えるイテレータ

- 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

## Enumerable モジュールで使えるイテレータ

- 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

## Enumerable モジュールで使えるイテレータ

- `sort`
  - 各要素を `<=>` メソッドで比較して、その結果を元にソートします

```
ary = [1, 3, 5, 2, 4].sort
p(ary)
```

[1, 2, 3, 4, 5]

## Enumerable モジュールで使えるイテレータ

- `sort_by`
  - 各要素のブロックの評価結果を `<=>` メソッドで比較して、その結果を元にソートします

```
ary = ["a", "C", "b", "D"].sort_by do |elem|
  elem.downcase
end
p(ary)
```

["a", "b", "C", "D"] # 小文字にしてからソート

## Enumerable モジュールで使えるイテレータ

- `inject`
  - まず初期値と最初の要素を引数にブロックを実行し、その後は前のブロックの実行結果と次の要素を引数に最後の要素まで順次ブロックを実行し、最後のブロックの実行結果を返します

```
val = [1, 2, 3, 4, 5].inject(0) do |sum, n|
  sum + n
end
```

```
p(val)
```

15 # 1 から 5 までの和

## 演習問題

- 「1 から 10 までの整数の積」を求めてください
  - 1 から 10 までの要素の配列は「`(1..10).to_a`」と書けます
  - 二通り以上思いっただけ書いてください
- 「1 から 20 までの整数から偶数だけを取り出した配列」を求めてください
- 「1 から 20 までの整数のなかの奇数の合計」を求めてください

## 演習問題

- 「パスカルの三角形を表示する」プログラムを書いてください

```
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
```

## 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/>

**Ruby 勉強会@関西**

<http://jp.rubyist.net/?KansaiWorkshop>

**rubyco ( るびこ ) の日記**

<http://d.hatena.ne.jp/rubyco/>

**ふえみにん日記**

<http://kazuhiko.tdiary.net/>