

# Ruby 初級者向けレッスン第 35 回

okkez @ Ruby 関西

2010 年 04 月 24 日

## 今回の内容

- Ruby 初級者向けレッスンの目的の説明
- Ruby の歩き方
- Rubyist の基礎知識

## 今回のゴール

- Ruby 初級者向けレッスンの目的を知る
- Ruby 関連の各種ドキュメントの在処を知る
- Ruby で困ったときの調べ方を知る

## Ruby 初級者向けレッスンの目的

- 対象と前提
- 目的と概要

## 対象と前提

- 小波ゼミの 3, 4 回生たち
- 初級者から中級者を目指す人
- 「たのしい Ruby 第三版」読了レベルを想定

## 目的と概要

- 初級者が中級者になるために必要な事を学ぶ
  - 扱う範囲は、基本的に組み込みライブラリや標準添付ライブラリまでです
  - RubyGems など簡単にインストール出来るライブラリやアプリケーションは紹介することがあります
  - Ruby らしいプログラムの書き方やイディオム等について積極的に学びます
  - 過去の初級者向けレッスンの内容と同じ内容が出てくることもあります

## Ruby の歩き方

以下の項目について順に紹介していきます。

- インストール
- チュートリアル
- 開発環境
- リファレンスマニュアル
- その他の情報源

## インストール

Ruby 勉強会に来るくらいなのだから Ruby のインストールはしているはずなので省略します。

## チュートリアル

### Ruby ではじめるプログラミング

- <http://jp.rubyist.net/magazine/?0002-FirstProgramming>
  - るびまの記事
  - プログラミングの初心者が対象
  - 他言語経験者には簡単過ぎるかも

### Ruby チュートリアル (沖ソフトウェア株式会社)

- <http://www.okisoft.co.jp/esc/ruby/tut.html>
  - JRuby のことも多少書いてある
  - Ruby 1.8.6 向け

## 開発環境

### Emacs, Meadow (ruby-mode)

#### GNU Emacs - GNU Project - Free Software Foundation (FSF)

<http://www.gnu.org/software/emacs/>

#### Meadow - Trac

<http://www.meadowy.org/meadow/>

言わずと知れた Emacs です。ruby-mode を使用して開発します。Ruby on Rails を使用して開発する場合は Rinari が便利です。

#### Rinari: Ruby on Rails Minor Mode for Emacs

<http://rinari.rubyforge.org/>

ちなみに、ruby-mode.el は Ruby の svn リポジトリに置いてあります。

- <http://svn.ruby-lang.org/repos/ruby/trunk/misc/ruby-mode.el>

#### okkez's dotfiles at master - GitHub

<http://github.com/okkez/dotfiles>

上記のように Emacs の設定ファイルを GitHub で公開している人がたくさんいるのでそういうものを参考にすると快適な開発環境を構築することができるでしょう。

### vim (ruby.vim)

#### welcome home : vim online

<http://www.vim.org/>

#### KaoriYa.net

<http://www.kaoriya.net/>

#### Vim-users.jp

<http://vim-users.jp/>

#### rails.vim - Ruby on Rails: easy file navigation, enhanced syntax highlighting, and more : vim online

[http://www.vim.org/scripts/script.php?script\\_id=1567](http://www.vim.org/scripts/script.php?script_id=1567)

いろいろなプログラミング言語にデフォルトで対応してるのが良いです。

### Eclipse (plugin)

#### Eclipse.org home

<http://www.eclipse.org/>

## FrontPage - EclipseWiki

<http://eclipsewiki.net/eclipse/>

## Aptana

<http://www.radrails.org/>

## Aptana (Rad Rails)

<http://www.apтана.com/>

プラグインによって拡張可能です。

## NetBeans (Ruby pack)

### ja: NetBeans 日本語サイト

<http://ja.netbeans.org/>

### ねこび~ん

<http://nekobean.net/>

Sun Microsystems が力を入れているオープンでフリーな IDE です。

Ruby pack が配布されているので、それをインストールしてください。インストールするだけでそこそこ使えるのが良いと思います。

- 最初から SCM 対応しています。
  - Mercurial
  - Subversion

## リファレンスマニュアル

古いのと新しいのがあります。古い方は以前は、RWiki で運用されていて誰でも編集出来るようになっていましたが、新しいリファレンスマニュアルの執筆が始まったため凍結されました。

ほとんどの内容は、新リファレンスマニュアルに移行されていますが、一部のドキュメントについては、旧版でしか参照できないものもあります。(移行自体はされているが参照できない)

- 旧: <http://www.ruby-lang.org/ja/man/>
- 新: <http://doc.okkez.net/>

旧版の方がページランクが高いため Google などでは検索結果として旧版が上に表示されますが、新版の方が内容は優れています。

## リファレンスを引くツール

- ri
- ReFe
- refe2, bitclust

## ri

ri は標準添付されている英語のリファレンスをコマンドラインから引くツールです。使い方は

```
$ ri --help
```

などで確認してください。

## ReFe

クラス名やメソッド名から Ruby のリファレンスマニュアルのエントリを引く、コマンドライン用のツールです。読みは「りふえ」。

- 基本的に自分でインストールする必要がある
- setup.rb を用いてインストールする方法と RubyGems を用いてインストールする方法がある
- One-Click Ruby Installer for Windows ( <http://rubyinstaller.rubyforge.org/wiki/wiki.pl> ) には同梱されている。
- 日本語の説明が読める
- 全てのクラスやメソッドのエントリが揃っているわけではない
- 検索機能が ri よりも強力
- 詳しい使い方は同梱されている README.ja を参照してください

## refe2, bitclust

Ruby リファレンスマニュアル刷新計画で作成されたリファレンスマニュアルをコマンドラインから引くことができるツール。

- refe2 は refe とほぼ同じように使うことができます。
- bitclust は Ruby リファレンスマニュアル刷新計画の中心を担うツールですが、リファレンスを引くのにも使用することができます。
- (bitclust の) ドキュメントがまだ揃っていません
- <http://doc.okkez.net/archives/> 以下で chm ファイルも配布しています。
- もっと簡単に使えるようにしたい

## その他の情報源

- メーリングリスト
- web サイト
- 書籍

## メーリングリスト

<http://www.ruby-lang.org/ja/community/mailling-lists/> を参照すれば良いと思います。

## その他のメーリングリスト

### Ruby 勉強会 @ 関西

Ruby 勉強会 @ 関西に関する情報が流れます。ほぼ、イベント告知用ですが、初級者向けレッスンに関する質問や要望も流していいと思います。

- <http://jp.rubyist.net/?RubyKansai>

## Web サイト

### Ruby

#### 公式サイト

<http://www.ruby-lang.org/>

#### Matz につき

<http://www.rubyist.net/~matz/>

#### Ruby コミッター一覧 - 世界線航跡蔵

<http://yugui.jp/articles/833>

#### Twitter / @yugui/ruby-committers

<http://twitter.com/yugui/ruby-committers>

## コミュニティを探す

### 日本 Ruby の会 公式 Wiki

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

### RubyKaigi 日記

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

### RubyKaigi 公式サイト

<http://www.rubykaigi.org/>

### Regional RubyKaigi

<http://regional.rubykaigi.org/>

### Rubyist SNS

<http://jp.rubyist.net/sns/>

## リソースを探す

### るりま Wiki (Ruby リファレンスマニュアル刷新計画)

<http://redmine.ruby-lang.org/wiki/rurema>

### Ruby Reference Manual (beta)

<http://doc.okkez.net/>

### Ruby リファレンスマニュアル - Ruby リファレンスマニュアル

<http://www.ruby-lang.org/ja/man/>

### Rubyist Magazine - るびま

<http://jp.rubyist.net/magazine/>

### Ruby チュートリアル (沖ソフトウェア株式会社)

<http://www.okisoft.co.jp/esc/ruby/tut.html>

## ソフトウェアを探す

### RubyForge: Welcome

<http://rubyforge.org/>

### RAA - Ruby Application Archive

<http://raa.ruby-lang.org/>

### GitHub

<https://github.com/>

## その他

### 技術系メーリングリストで質問するときのパターン・ランゲージ

<http://www.hyuki.com/writing/techask.html>

## 書籍

### 初学者向け

- たのしい Ruby 第3版 ISBN:4797357401
- 初めてのプログラミング ISBN:4873112923
- Ruby 1 はじめてのプログラミング ISBN:9784798117997
- Ruby 2 さまざまなデータとアルゴリズム ISBN:9784798118000
- Ruby 3 オブジェクト指向とはじめての設計 ISBN:9784798118017

## 初級者向け

- 初めての Ruby ISBN:9784873113678
- Ruby で作る奇妙なプログラミング言語 Esoteric Language ISBN:9784839927844
- Ruby レシピブック 第二版 ISBN:4797340045
- Ruby 逆引きハンドブック ISBN:9784863540224
- Ruby 逆引きレシピ ISBN:9784798119885

## 中級者向け

- Ruby クックブック ISBN:4873113245
- Rubyist Magazine 出張版 ISBN:4839923205
- Rubyist Magazine 出張版 Ruby on Windows ISBN:4839926689
- Ruby スクリプティングテクニック ISBN:9784873113562
- Ruby ベストプラクティス ISBN:4873114454
- リファクタリング:Ruby エディション ISBN:4048678841

## 上級者向け

- プログラミング Ruby 第二版
  - 言語編 ISBN:4274066428
  - ライブラリ編 ISBN:4274066436
- The Ruby Way 第二版 ISBN:9784798115337
- 実践 Rails ISBN:9784873113869

他にも色々あります。Amazonなどで”Ruby”で検索してみてください。

## Stack Stock Books - Ruby

<http://stack.nayutaya.jp/tag/ruby>

## 情報収集 (勉強) の方法

- リファレンスマニュアルを通読する
  - よく使うクラスに関しては通読しておくが良い
  - String, Regexp, Array, Enumerable, Kernel, Object など
- 検索エンジンで検索する

- エラーメッセージで検索してエラーの原因を探る
- あるクラスやメソッドの名前で検索して使用例を探す
- RAA, RubyForge, GitHub で探す
  - 特定の目的に使えるライブラリを探す
  - GitHub は使っている人が多い
- ML で質問する
  - 検索のためのキーワードがわからない場合
  - 自分ではどうしようもない場合
- 掲示板やフォーラムで質問する
  - ML で質問する場合と同様
- ブログで悩んでいる過程を公開する
  - 自分が悩んでいる過程をありのままに公開する
  - 同じことで悩んで(る|た)人がコメントくれるかも
  - すごい人にコメントもらえるかも
- 勉強会で発表する
  - 発表する事によって準備時に勉強しなければならなくなる
  - 発表後の質疑応答でいい質問が飛んでくるかもしれない
  - 懇親会でいろいろな人と話しやすくなる

## Rubyist の基礎知識

- クラスとは
- 各種メソッドの使い分け
- アクセス制御
- 各種変数の使い分け
- モジュール

以下の説明では Ruby ではこういうものがある、という説明をしていくのでよくわからなかったところは、質問時間に質問したり、家に帰ってから調べるなりしてください。

## クラスとは

Ruby ではクラスも Class クラスのインスタンスです。なので以下のような記述でもクラス定義出来ます。

```
01: # -*- coding: utf-8 -*-
02: # ちなみに new の引数には継承したい親クラスを指定します
03: Book = Class.new{
04:   attr_reader :isbn, :title, :author, :price
05:   define_method(:initialize){|isbn,title,author,price|
06:     @isbn = isbn
07:     @title = title
08:     @author = author
09:     @price = price
10:   }
11:   define_method(:reading){
12:     puts '未来の本なので読み上げ機能が付加されています'
13:   }
14:   define_method(:spec){
15:     puts "#{isbn}, #{title}, #{author}, #{price}円"
16:   }
17: }
18: # 呼び出しも普通にできます
19: book = Book.new('9784873113678', '初めての Ruby', 'Yugui', 2200)
20: book.reading
21: book.spec
```

先ほどのものを普通の Ruby の書き方で書くとこうなります。

```
01: # -*- coding: utf-8 -*-
02: class Book
03:   attr_reader :isbn, :title, :author, :price
04:
05:   def initialize(isbn, title, author, price)
06:     @isbn = isbn
07:     @title = title
08:     @author = author
09:     @price = price
10:   end
11:
12:   def reading
13:     puts '未来の本なので読み上げ機能が付加されています'
14:   end
15:
16:   def spec
```

```

17:     puts "#{isbn}, #{title}, #{author}, #{price}円"
18:   end
19: end
20:
21: book = Book.new('9784873113678', '初めての Ruby', 'Yugui', 2200)
22: book.reading
23: book.spec

```

- どちらがわかりやすいかは一目瞭然ですね
- def ~ end を受け付けられない体質の友人が居たら前者の書き方を教えてあげてください

前者の書き方は例外クラスを一行で定義するときに使われることがあります。

```

01: module Foo
02:   BarError = Class.new(StandardError)
03: end

```

よくあるクラスの説明では、「設計書」や「たい焼きの型」と例えられることがあります。

## インスタンスメソッド

あるインスタンスをレシーバとするメソッドです。例えば、下の Taiyaki#yakitate?メソッドはインスタンスメソッドです。

```

01: # -*- coding: utf-8 -*-
02: class Taiyaki
03:   attr_reader :created_at, :cream
04:   def initialize(cream)
05:     @created_at = Time.now
06:     @cream = cream
07:   end
08:   # 焼き上がってから 5 分以内なら焼きたて
09:   def yakitate?
10:     (Time.now.to_i - created_at.to_i) <= 300
11:   end
12: end
13: # 実行
14: obj = Taiyaki.new('つぶあん')
15: sleep(10)
16: p obj.yakitate? #=> true
17: sleep(300)
18: p obj.yakitate? #=> false

```

## クラスメソッド

クラスそのものをレシーバとするメソッドです。例えば、下の `Taiyaki.prepared?`, `Taiyaki.heating` はクラスメソッドです。

```
01: # -*- coding: utf-8 -*-
02: class Taiyaki
03:   @@temperature = 0
04:   # 型の温度が 150 度より高ければ準備 OK
05:   def self.prepared?
06:     @@temperature > 150
07:   end
08:   def self.heating
09:     @@temperature += 10
10:   end
11: end
12: # 実行
13: p Taiyaki.prepared? #=> false
14: 16.times{ Taiyaki.heating }
15: p Taiyaki.prepared? #=> true
```

## 関数的メソッド

- レシーバがないメソッド
- Kernel モジュールの `private instance method` として定義されている
- Object クラスが Kernel モジュールを `include` しているので使えると覚えておけばよい
- クラス定義内でレシーバを書かずに、メソッド呼出しできるのは `self` というレシーバが省略されているから

## アクセス制御

### `public`

メソッドをインスタンスメソッドとして使えるように公開する。

### `private`

メソッドをクラスの内部だけで使えるようにする。レシーバを指定して呼び出せないようにする

### `protected`

メソッドをクラスの内部から使えるようにする。同一クラス内ではインスタンスメソッドとしても使えるようにする。

例:

```

01: # -*- coding: utf-8 -*-
02: class AccTest
03:   def pub_method
04:     puts 'this is a public method'
05:   end
06:   def priv_method
07:     puts 'this is a private method'
08:   end
09:   private :priv_method
10: end
11: # 呼び出してみる
12: obj = AccTest.new
13: obj.pub_method
14: obj.priv_method
15:
16: # ~> -:14: private method 'priv_method' called for #<AccTest:0x7f52f415d2d8> (NoMethodError)
17: # >> this is a public method

```

例:

```

01: # -*- coding: utf-8 -*-
02: class Point
03:   attr_accessor :x, :y # アクセスメソッドを定義する
04:   protected :x=, :y= # x=, y= を protected にする
05:   def initialize(x = 0.0, y = 0.0)
06:     @x = x
07:     @y = y
08:   end
09:   def swap(other)
10:     x_tmp = @x
11:     y_tmp = @y
12:     @x = other.x
13:     @y = other.y
14:     other.x = x_tmp
15:     other.y = y_tmp
16:   end
17: end
18:
19: obj1 = Point.new
20: obj2 = Point.new(1.0, 2.0)
21:
22: obj1 # => #<Point:0x7f5be0e1a5a8 @x=0.0, @y=0.0>
23: obj2 # => #<Point:0x7f5be0e1a378 @x=1.0, @y=2.0>
24:

```

```
25: obj1.swap(obj2)
26:
27: obj1 # => #<Point:0x7f5be0e1a5a8 @x=1.0, @y=2.0>
28: obj2 # => #<Point:0x7f5be0e1a378 @x=0.0, @y=0.0>
29:
30: obj1.x      # => 1.0
31: obj1.x = 10.0
32:
33: # ~> -:31: protected method 'x=' called for #<Point:0x7f5be0e1a5a8 @x=1.0, @y=2.0> (NoMethodError)
```

Ruby の場合は厳密にアクセス制御してもリフレクションを使うことによって比較的簡単にアクセス制御を突破出来るので、もっと別の部分に気を配った方が良いでしょう。

## 各種変数について

### ローカル変数

先頭がアルファベットの小文字が”\_”ではじまるもの。最もよく使うが、最も有効範囲が狭い。

### グローバル変数

先頭が”\$”で始まるもの。最も有効範囲が広いが、ユーザーが定義して使うのは嫌われる。定義済みのものはリファレンスマニュアル参照。

### クラス変数

先頭が”@@”ではじまるもの。そのクラスの全てのインスタンスで共有できる変数。あまり使わない方が良い。

### インスタンス変数

先頭が”@”ではじまるもの。クラスを作るとほぼ必ず使う。

## モジュールとクラスの違い

- クラスは継承することができるが、モジュールは継承することができない
- クラスはインスタンス化できるが、モジュールはインスタンス化できない

## モジュールの使いどころ/使われどころ

- よく似た処理をまとめるのに使われる
  - Enumerable, Comparable など
- 名前空間の代わり
  - Net::{HTTP,HTTPS,FTP,SMTP,IMAP} など

## 演習

### るりまの成果物をダウンロードしてインストールしてみよう

<http://doc.okkez.net/> からるりまの成果物をダウンロードして、インストールしてみてください。

- 201003 の chm をダウンロードして使ってみる
  - Windows Vista の人は起動時に出てくる警告のチェックボックスを外すといいそうです
- 201003 の tar.bz2, tar.gz, zip を使用してみる。(中身は同じです)
  - Windows の人は展開に時間がかかります
  - Windows の人は server.exe を実行してみましょう
- 新しい refe も使ってみてください

### ローマ数字 Part1

Integer クラスを拡張して、ローマ数字を返すメソッドを作ってみましょう。

初期の頃のローマ数字には減算則はありませんでした。つまり、「IX」で9を表したりせず、素直に、大きい数字の次に小さい数字を順に記述して「VIII」をいうように書いていました。

# 出力例

```
puts 4.to_old_roman #=> IIII
puts 115.to_old_roman #=> CXV
```

- メソッド名は to\_old\_roman
- 変換規則は以下の通り

1 = I, 5 = V, 10 = X, 50 = L, 100 = C, 500 = D, 1000 = M

### ローマ数字 Part2

新しいタイプのローマ数字の文字列を返すメソッドを作りましょう。

# 出力例

```
4.to_roman #=> IV
999.to_roman #=> IM
```

- メソッド名は to\_roman
- 考え方は Part1 と同じで大丈夫です。

## まとめ

- Ruby の歩き方を学びました
- Ruby に関する情報のある場所を学びました
- どんどんプログラムを書きましょう
- 何かあったらブログを書きましょう
- Ruby ではクラスも Class クラスのインスタンス

## 参考文献

### 初めてのプログラミング

<http://www.oreilly.co.jp/books/4873112923/>

### プログラミング Ruby 第 2 版 言語編

<http://ssl.ohmsha.co.jp/cgi-bin/menu.cgi?ISBN=4-274-06642-8>

### たのしい Ruby 第 3 版

<http://www.sbcjr.jp/books/products/detail.asp?sku=4797357400>

## 今後の情報源

### 公式 Web サイト

<http://www.ruby-lang.org/>

### Ruby リファレンスマニュアル刷新計画

<http://doc.loveruby.net/>

### 日本 Ruby の会

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

### Rubyist Magazine

<http://jp.rubyist.net/magazine/>

### okkez weblog

<http://typo.okkez.net/>