

Ruby 講義

第4回 Ruby入門

Kuniaki IGARASHI

2012.5.1 at 一橋大学

**社会科学における情報技術とコンテンツ作成III
(ニフティ株式会社寄附講義)**

今日の資料 <http://bit.ly/ruby0501>

○ 剰余金の配当に関するお知らせ

○ ニフティ、「@nifty EMOBILE LTE 定額にねんプラン」の提供を開...

○ 「@nifty温泉」で「母の日 全国一斉！100のありがとう風呂」特設サイト公...

○ 「スマブレ！」のサービス停止について

○ ニフティとサンリオウェブ、iOS向けアプリ「Hello Kitty Worl...

○ 平成24年3月期 決算短信

○ 特別損失の計上に関するお知らせ

○ 「シュフモ」登録会員数150万人を突破、「2012年 主婦の全国節電調査（冬季...

ニフティとなら、きっとかなう。
With Us, You Can.

ニフティ株式会社

HOME

NIFTY

ニュースリリース

アット・ニフティ

楽しいサービスがいっぱい

@nifty

アクセスマップ

大森から西新宿へ移転いたしました

@nifty Web募金

東日本大震災復興支援
募金受付中



2012年4月25日 IR 特別損失の計上に関するお知らせ

2012年4月25日 IR 剰余金の配当に関するお知らせ

2012年4月25日 IR 平成24年3月期決算短信

2012年4月11日 IR ニフティ、「@nifty EMOBILE LTE 定額にねんプラン」の提供を開始

2012年4月11日 IR ニフティとサンリオウェブ、iOS向けアプリ「Hello Kitty World」を台湾で提供開始

2012年4月10日 お知らせ 「@nifty温泉」で「母の日 全国一斉！100のありがとう風呂」特設サイト公開

講師

五十嵐邦明

株式会社万葉

エンジニア



EM...
編集後記とま...

GARASHI

.9.25 at 高専カンファ

Teaching Assistant 濱崎 健吾
クックパッド株式会社 エンジニア



講義資料置き場

講義資料置き場をつくりました。
過去の資料がDLできます。

<https://github.com/hitotsubashi-ruby/lecture2012>
or

<http://bit.ly/ruby-lecture>

雑談・質問用facebookグループ

facebookグループを作りました

<https://www.facebook.com/groups/hitotsubashi.rb>

- 加入/非加入は自由です
- 加入/非加入は成績に関係しません
- 参加者一覧は公開されます
- 書き込みは参加者のみ見えます
- 希望者はアクセスして参加申請してください
- 雑談、質問、議論など何でも気にせずどうぞ～
- 質問に答えられる人は答えてあげてください
- 講師陣もお答えします
- 入ったら軽く自己紹介おねがいします



教科書 p.4 ~ p.32

やること

前回のおさらい

shellの話

変数

条件判断、繰り返し

メソッドの作成

別のファイルを読み込む(require文)

```
print("Hello, Ruby.\n")
```

オブジェクト (Stringオブジェクト)

```
print("Hello, Ruby.\n")
```

メソッド

引数

メソッド：手続き、命令

引数：メソッドに渡すデータ

(ここでは"Hello, Ruby.\n"オブジェクトが引数)

計算

puts 1 + 2

puts 2 - 3

puts 5 * 10

puts 100 / 4

puts 2**32

puts 10 / 3

puts 10 / 0

× は *

÷ は /

累乗は **

メソッド puts

演算子 + とか

整数オブジェクト

1 とか

(Fixnumオブジェクト)

計算するときの注意

整数オブジェクト同士の計算は整数オブジェクトになる

> puts 10/3

$\Rightarrow 3 \leftarrow 3.33\dots$ ではない

**小数で求めたい場合は両方もしくは片方を
小数(Float)オブジェクト(例: 10.0)にする**

```
> puts 10.0/3.0
```

$\Rightarrow 3.3333333333333335$

小数(Float)オブジェクトの計算は不正確な場合がある

エラーメッセージは お得な情報を教えてくれる

Rubyが教えてくれたエラーメッセージ

```
helloruby.rb:2:in `<main>': undefined  
method `prin' for main:Object  
(NoMethodError)
```

日本語訳

helloruby.rb というファイルの 2 行目で
prinなんてメソッドはないので
そんなメソッドなかとよエラー が起きたよ

文字コードと マジックコメント

日本語を含むコードを書くとき

- ファイルをUTF-8で保存する。
- 1行目にマジックコメントを書く
(そのファイルの文字コードをRubyへ教えるため)

例) `# encoding: utf-8`

`# encoding: utf-8` ←マジックコメント

```
print("ちはやふるかみよもきかずたつたがわ\n")
```

```
print("からくれないにみずくくるとは\n")
```

Ruby実行環境

shellとeditorを使う

Ubuntu

- shell: 端末(terminal)
- editor: テキストエディター(gedit)

Windows

- shell: Command Prompt with Ruby and Rails
- editor: intype

Rubyコードを実行する方法

- **irb から1行ずつ実行**
- **ファイル(.rb) にコードを書いて実行**

irbから1行ずつ実行

```
$ irb
```

shell

```
1.9.3p125 :001 > puts 1+2+3
```

```
6
```

```
=> nil
```

```
1.9.3p125 :002 > exit
```

irb

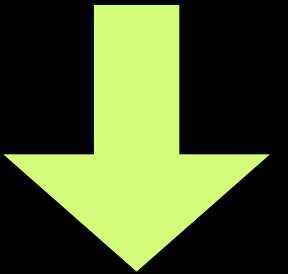
```
$ pwd
```

```
/Users/igarashi
```

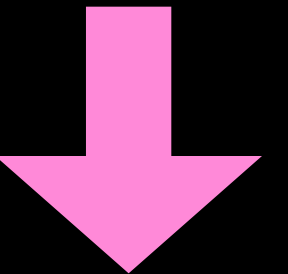
```
$
```

shell

irb



exit



ファイルにコードを書いて実行

エディタでhelloruby.rbを作って保存

```
print("Hello, Ruby.\n")
```

shellでhelloruby.rbを実行

```
$ ruby helloruby.rb
```

```
Hello, Ruby.
```

shellでRubyコードが書かれたファイルを実行するには
ruby ファイル名

**ここから
今週の内容!!
shellの話を
もう少し解説します**

**shellはOSと話をするための世界です。
良く使うコマンドを簡単に説明します。**

ls : ファイル一覧を見る

cd : フォルダ移動

mkdir : フォルダ作成

pwd : 今いるフォルダ名を表示

cat : ファイルの中身を表示

ls : ファイル一覧を見る

ファイル一覧を表示します。フォルダーも見れます。

```
$ ls
```

※コマンドは \$ 始まりで書くことが多いです。この\$は始まりの印（プロンプトと言います）なので打たなくて大丈夫です。ls とだけ打ちます。

オプション(スペースに続けて -a や -l を指定可能)

\$ ls -a (ファイル名が.で始まる隠しファイルも表示)

\$ ls -l (ファイル作成日など詳細情報を表示)

cd : フォルダ移動

フォルダを移動するコマンドです。

```
$ cd フォルダ名
```

親フォルダへ移動する場合は .. を使います。

```
$ cd ..
```

/ で区切って孫フォルダへ一気に移動もできます。

```
$ cd src/Ruby1
```

=> srcフォルダの中にあるRuby1フォルダへ移動

mkdir : フォルダ作成

フォルダを作るコマンドです。

```
$ mkdir フォルダ名
```

ちなみに消すのは rmdir コマンドです。

```
$ rmdir フォルダ名
```

フォルダの中が空でないと rmdir では削除できません。
フォルダの中にファイルなどがあるのに消したい場合は rm
-rf コマンドで削除できます。※削除したファイルは復元で
きないので注意して使ってください！！

★危険★

```
$ rm -rf フォルダ名
```



pwd : 今いるフォルダ名を表示

今いるフォルダを表示します。

```
$ pwd  
/home/dev/
```

フォルダが階層構造になっているときは / で区切られます。

**Rubyのコードが
かかれたファイルを
実行するためには**

Rubyコードを実行する方法

1. XXX.rb があるフォルダへ移動する
cdコマンドを使います。

例) `$ cd src/0501`

2. `ruby XXX.rb` で実行します。

例) `$ ruby XXX.rb`

もしも以下のエラーが出たらlsコマンドでファイル有無を確認します。

`ruby: No such file or directory -- XXX.rb (LoadError)`

shellを起動した時の始点フォルダ

Ubuntu : /home/ユーザー名/

Windows : /C/Sites (C:¥Sites)

例えば上記の下に src みたいなフォルダを作って .rb ファイルを整理するのがオススメです。(VM Ubuntuでは作成済み)

```
$ mkdir src  
$ cd src  
$ mkdir 0501  
$ cd 0501  
$ ruby XXX.rb
```


では、引き続き
Rubyのお話です

変数

教科書 p.21

オブジェクト へのラベル・荷札

変数 = オブジェクト

変数にオブジェクトを代入する

```
name = "igarashi"
```

(変数nameに"igarashi"オブジェクトを代入)

```
puts name  
=> "igarashi"
```

変数

教科書 p.21-23

では、この場合は
どうでしょうか？

サンプルコード

```
a = "abc"
```

```
b = a
```

```
a.upcase!
```

```
puts a
```

```
puts b
```

※**upcase!** は
Stringオブジェクトを
大文字にするメソッド

a は **"ABC"** になりますが、
b は どうなるでしょう？

```
a = "abc"
```

```
b = a
```

```
a.upcase!
```

```
puts a  
puts b
```

aはオブジェクト
"abc"を示
す変数

bもaと同じ
"abc"を示す
変数

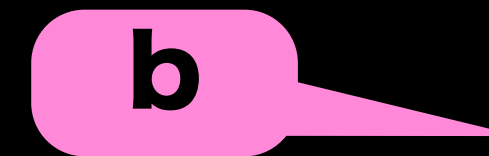
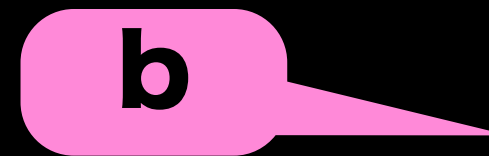
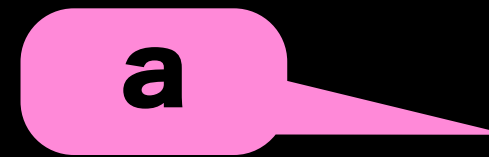
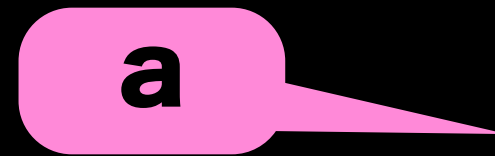
aの指すオブ
ジェクトを大
文字にする

a => "ABC"

b => "ABC"

変数

オブジェクト



さっきと似てるけどちょっと違うコード

```
a = "abc"
```

```
b = "abc"
```

```
a.upcase!
```

```
puts a  
puts b
```

さっきは→
こうでした

```
a = "abc"
```

```
b = a
```

```
a.upcase!
```

```
puts a  
puts b
```

```
a = "abc"
```

```
b = "abc"
```

```
a.upcase!
```

```
puts a
```

```
puts b
```

aはオブジェクト
"abc"を示
す変数

bは別のオブジェ
クト"abc"を指
す

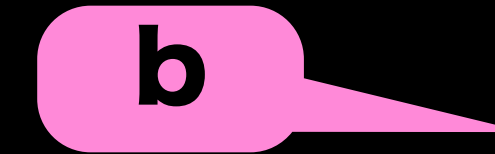
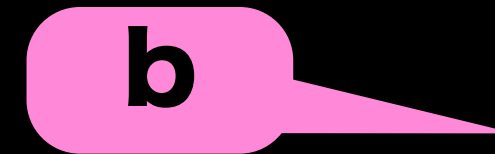
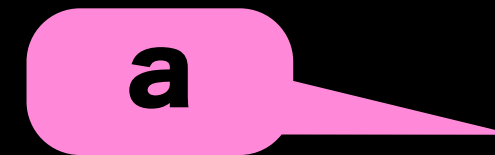
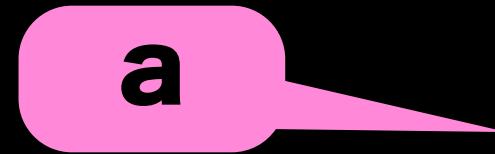
aの指すオブ
ジェクトを大
文字にする

a => "ABC"

b => "abc"

変数

オブジェクト



名付け重要

変数名は分かりやすい名前にしよう

良い例

```
width = 20  
height = 3  
area =  
    width * height
```

悪い例

```
a1 = 20  
a2 = 3  
a3 =  
    a1 * a3
```

コメント文

教科書 p.23-24

条件判断: if文

教科書 p.25-27

条件判断 if 文

※教科書には `then` が書いてありますが、省略可能です。
普通は省略します。私は書いたことないです。

if 条件

条件が成り立ったときに実行したい処理

end

**条件には値が `true`(真) または `false` (偽)
となる式を書くことが一般的**

条件判断 == 演算子

```
x = 3 - 2  
if x == 1  
    puts "x is 1"  
end
```

x が 1 と同じか判断し
x が 1 の時に
puts が実行されます。

== は左辺(x)と右辺(1)が同じかどうか調べて、
同じならば true、異なれば false になります。
比較は == です。= だと代入になります。
ちなみに、異なるかを判断する != もあります。

条件判断 if - else - end

if 条件

条件が成立した時に実行したい処理

else

条件が不成立の時に実行したい処理

end

条件がfalseの時に実行したい処理を書くこともできます。

if 文は後ろにも書ける

条件成立時に実行したい処理 if 条件

```
if x == 1  
  puts "x is 1"  
end
```

左の文は以下のように1行で書くこともできます。

```
puts "x is 1" if x == 1
```

1行で書ける条件

- ・ 実行したい処理が1行だけのとき
- ・ else節を書かないとき

if文の演習


1. コードを実行した際に true が表示されるように空欄を埋めてコードを完成させて実行してください。

```
x = 2
if                     
  puts 'true'
end
```

2. x が "GW" という文字列のときは Yeah! 、それ以外のときは sigh... と表示させるコードを書いて実行させてください。

インデント(字下げ)

```
if x == 1  
    puts "x is 1"  
end
```



例えばif文中など、こういう風に先頭にスペースを入れて書くことをインデントするといいます。

プログラムの実行には不要なものですが、

絶対に入れてください！

無いと人が読めないの・・・

ちなみにスペースの個数には流派がありますが、2個が主流のようです。

繰り返し

while文

timesメソッド

教科書 p.27-28

メソッドの作成

教科書 p.29

メソッドの定義、呼び出し

```
def メソッド名  
  メソッドで実行したい処理  
end
```

メソッド定義
には **def** を
使う

```
def hello  
  print "Hello, Ruby.\n"  
end
```

定義

```
hello()
```

呼び出し

メソッド呼び出しはメソッド名に()をつけます。
そして、この()は省略可能です。(曖昧にならない限り)

別のファイルを

取り込む

教科書 p.30-31


requireメソッドの演習

p.31のList1.8,1.9を実行してください。

```
def hello  
  print "Hello, Ruby.\n"  
end
```

List1.8
hello.rb

```
require "./hello"  
hello()
```



List1.9
use_hello.rb

教科書は require "hello" になっていますが、Ruby1.9.2以降だとエラーになるので "./hello" としてください。 ./ は (shellで)今いるフォルダの意味です。

```
$ ruby use_hello.rb
```

実行例

今日の

まとめ

shell のコマンド

ls : ファイル一覧を見る

cd : フォルダ移動

mkdir : フォルダ作成

pwd : 今いるフォルダ名を表示

cat : ファイルの中身を表示

変数

教科書 p.21

オブジェクト へのラベル・荷札

変数 = オブジェクト

変数にオブジェクトを代入する

```
name = "igarashi"
```

(変数nameに"igarashi"オブジェクトを代入)

```
puts name  
=> "igarashi"
```

条件判断

if - end

if 条件

条件が成立した時に実行したい処理

end

if - else - end

if 条件

条件が成立した時に実行したい処理

else

条件が不成立の時に実行したい処理

end

比較演算子 ==, !=

メソッドの定義、呼び出し

```
def メソッド名  
  メソッドで実行したい処理  
end
```

メソッド定義
には **def** を
使う

```
def hello  
  print "Hello, Ruby.\n"  
end
```

定義

```
hello()
```

呼び出し

メソッド呼び出しはメソッド名に()をつけます。
そして、この()は省略可能です。(曖昧にならない限り)

別のファイルを 取り込む

```
require "./hello"
```

実行フォルダにある hello.rb を取り込む

対応する教科書のページ

変数 p.21～23

コメント p.23～24

条件判断・繰り返し p.24～28

メソッド p.29

別のファイルの読み込み(require)

p.30～31

if文の演習 解答

1.

```
x = 2
if x == 2
  puts 'true'
end
```

2. x が "GW" という文字列のときは Yeah! 、それ以外のときは sigh... と表示させるコードを書いて実行させてください。

```
x = "GW"
if x == "GW"
  puts "Yeah!"
else
  puts "sigh..."
end
```

講義資料置き場

講義資料置き場をつくりました。
過去の資料がDLできます。

<https://github.com/hitotsubashi-ruby/lecture2012>
or

<http://bit.ly/ruby-lecture>

雑談・質問用facebookグループ

濱崎さん作のfacebookグループができました

<https://www.facebook.com/groups/hitotsubashi.rb>

- 加入/非加入は自由です
- 加入/非加入は成績に関係しません
- 参加者一覧は公開されます
- 書き込みは参加者のみ見えます
- 希望者はアクセスして参加申請してください
- 雑談、質問、議論など何でも気にせずどうぞ～
- 質問に答えられる人は答えてあげてください
- 講師陣もお答えします
- 入ったら軽く自己紹介おねがいします