

# Ruby 講義

## 第3回 irb, shell, 四則演算

五十嵐邦明

twitter : igaiga555

<http://www.facebook.com/kuniaki.igarashi>



2013.4.25 at 一橋大学  
ニフティ株式会社寄附講義  
社会科学における情報技術と  
コンテンツ作成III

○ 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ーブ、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

○ 「シユフモ」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.

HOME

会社概要

会社紹介

社会・環境活動

ニュースリリース

IR情報

採用情

専門情報

労働環境や働く環境をさらに



アット・ニフティ  
楽しいサービスがいっぱい



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



# ニフティ株式会社

## 寄付請求

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19日 プレス フテナモ「@nifty EMOBILE LTE 定額にねんプラン」の提供を開始

2012年4月10日 プレス フテナモ「@nifty EMOBILE LTE 定額にねんプラン」を台湾で提供

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

# 五十嵐邦明 講師 株式会社万葉



twitter: igaiga555

<https://github.com/igaiga/>

<http://www.facebook.com/kuniaki.igarashi>

# 濱崎 健吾

Teaching Assistant  
fluxflex, inc(米国法人)



twitter: hmsk

<https://github.com/hmsk/>

<http://www.facebook.com/hamachang>

やること

irb でRubyプログラムを実行する

用語の説明：オブジェクト、メソッド、引数

四則演算

.rb ファイルでRubyプログラムを実行する

shell

日本語、文字コード、マジックコメント

# Ruby環境構築

前回の講義時の環境構築をまだ済ませてない方は、以下の資料に沿ってRubyを実行する環境を作ってください。  
RailsInstallerとエディタがセットアップできればOKです。(後半のVirtualBox Ubuntu Image 以降はオプションです。)

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

# 3つの世界

**Ruby(irb)**

Ruby語が通じる世界  
1行ずつコードを実行

**Ruby(ファイル)**

Ruby語が通じる世界  
ファイルにコードを書いて実行

**Shell**

OS語が通じる世界  
ターミナルの中

言葉の解説：コード = プログラム です。  
コードって言うとプロっぽくてかっこいいです。

# 3つの世界

**Ruby(irb)**

Ruby語が通じる世界  
1行ずつコードを実行

**Ruby(ファイル)**

Ruby語が通じる世界  
ファイルにコードを書いて実行

**Shell**

OS語が通じる世界  
ターミナルの中

まずはirbを使ってRubyのコードを書いてみましょう。

# irb (Interactive RuByの略)

irb はrubyコードを1行ずつ実行する環境です。

起動方法

Mac : ターミナル

Win : "Command Prompt with Ruby and Rails"

Linux(Ubuntu) : 端末(※分からぬ場合は次のページ参照)

※ ↑ どれも今後はターミナルと呼びます。 (ターミナルの和訳が端末)

起動して以下のように入力してEnterを押してください。

\$ irb

※先頭の \$ はターミナルであることを表すマークです。  
(\$と、その次のスペースは入力不要です。 irbと打てばOK。)  
今後、ターミナルで打つコマンドは同じ書式  
(紫の背景色、 \$ マーク)で書きます。

こんな風に表示が出ればOKです。

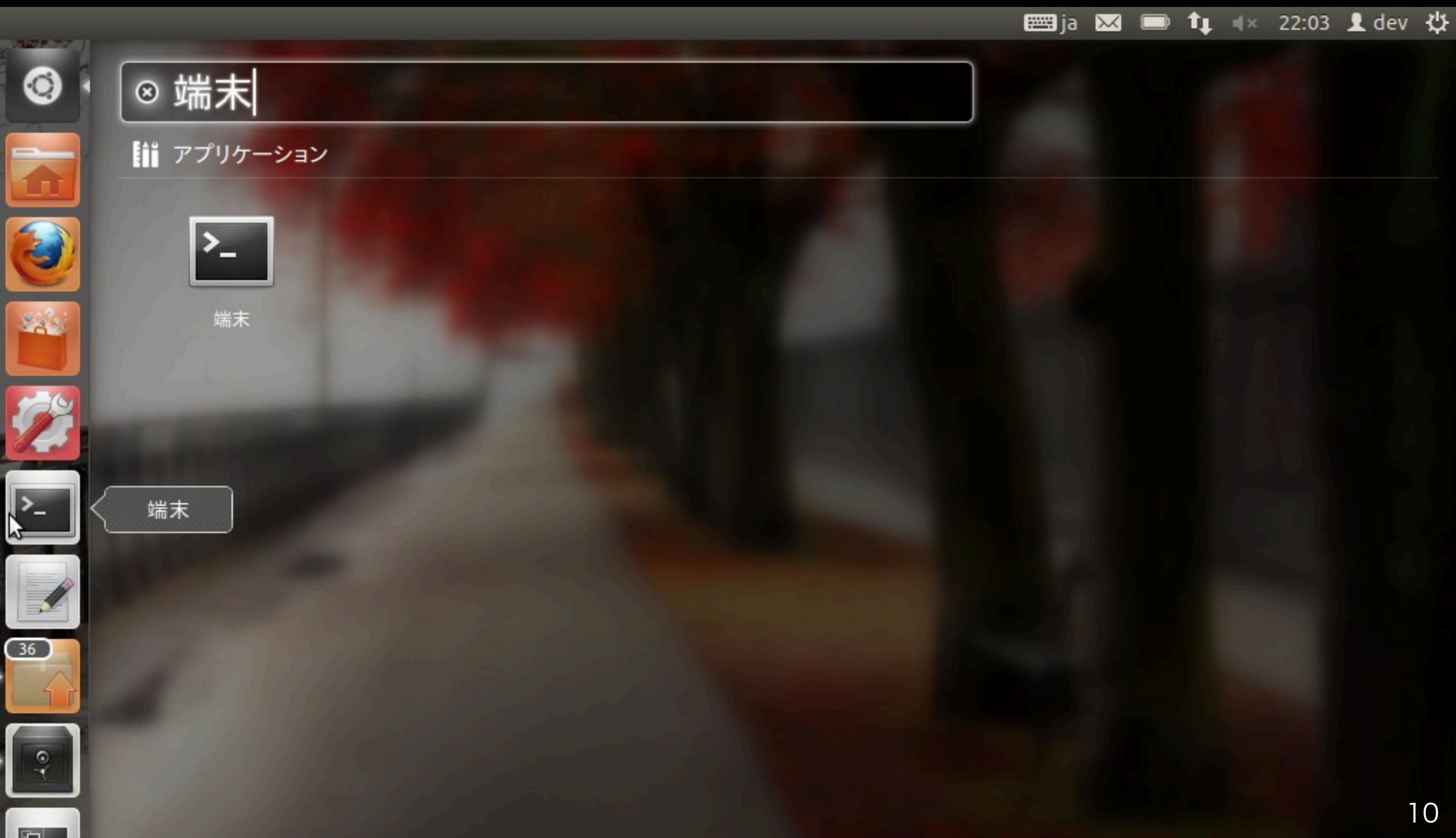
2.0.0p0 :001 >

※irbを終了させるには exit と打ちます。



# Linux(Ubuntu) 端末起動

1.左メニューの一番上のボタンを押し、検索ボックスに「端末」と打ちます



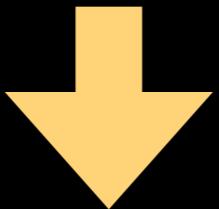
# 画面に Hello world! と表示するコードを実行

irbで以下のRubyコードを打って、Enterを押して実行します。

```
puts("Hello world!")
```

※緑の背景色はRubyのコードを表します。

```
2.0.0-p0 :001 > puts("Hello world!") ↵enter
```

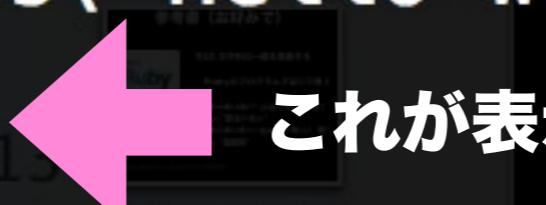


```
2.0.0-p0 :001 > puts("Hello world!")
```

```
Hello world!
```

```
=> nil
```

```
2.0.0-p0 :002 > _
```



これが表示されれば成功です。

※irbで画面に表示される行頭の2.0.0-p0はRubyのバージョンです。  
環境によって異なります。

# 画面に文字を表示するコード

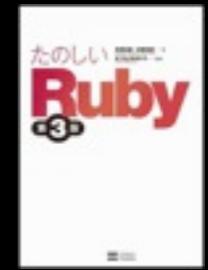
画面に文字を表示するコードには、  
さきほどの `puts` のほかに、  
`print`, `p` というメソッドもあります。

`puts("Hello world!")` 表示後、改行

`print("Hello world!")` 表示後、改行しない  
改行する場合は `\n` (Winの場合は`\r\n`) を入れます。  
`print("Hello\nworld!\n")`

`p("Hello world!")` デバッグ(=不具合修正)用  
この先の講義でよく使います

# よく使う 用語の説明



教科書  
p.8,9

`puts("Hello, World!")`

オブジェクト

`String(文字列)` オブジェクト

# `puts("Hello, World!")`



教科書  
p.8,9

## 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

"Hello, World!" はString(文字列)オブジェクトであり、この場合はメソッドへ渡しているので引数でもあります。

また、引数の前後についているカッコ () は(原則)省略可能で、() の代わりにスペースで書くこともできます。  
つまり、以下の2文は同じ意味です。

```
puts("Hello, World!")
puts "Hello, World!"
```

次は計算を  
してみましょう

# 以下の計算をしてみましょう

```
puts 1 + 2  
puts 2 - 3  
puts 5 * 10  
puts 100 / 4  
puts 2**32  
puts 10/3  
puts 10/0
```

※ () を省略した書き方で  
書いています。

+ : 足し算  
- : 引き算  
\* : 掛け算  
/ : 割り算  
\*\* : 累乗

logとかsinとかもあります。  
知りたい方はこちら。

# 言葉の説明

**puts 1 + 2**

メソッド

**puts 2 - 3**

puts

**puts 5 \* 10**

演算子

**puts 100 / 4**

+とか -とか

**puts 2\*\*32**

オブジェクト

**puts 10 / 3**

1とか2とか

**puts 10 / 0**

整数オブジェクト

(Fixnumオブジェクト)

# 計算するときの注意

`puts 1 + 2`

`puts 2 - 3`

`puts 5 * 10`

`puts 100 / 4`

`puts 2**3`

`puts 10 / 3 => 3`

`puts 10 / 0`

ところで、これ、  
結果が変だと  
思いませんか？

# 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

=> 3 ← 3.33... ではない

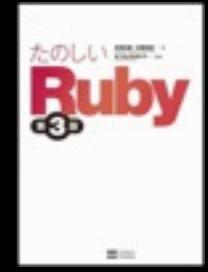
小数で求めたい場合は両方もしくは片方を  
小数(Float)オブジェクト(例：10.0)にする

> puts 10.0/3.0

=> 3.333333333333333

> puts 10/3.0

=> 3.333333333333333



# 数値オブジェクト

Rubyで数値を扱う場合

2：整数オブジェクト

(Fixnumオブジェクト)

3.14：小数（浮動小数点数）オブジェクト

(Floatオブジェクト)

ほかにもあります。そのうちでてきます。

# 計算するときの注意

`puts 1 + 2`

`puts 2 - 3`

`puts 5 * 10`

`puts 100 / 4`

`puts 2**3`

`puts 10 / 3 => 3`

**`puts 10 / 0`**

ところで、これ、  
何言ってんですか？

**ZeroDivisionError:**  
**divided by 0**  
**from (irb):2:in `/'**  
**from (irb):2**

# エラーメッセージ

正しくないプログラムを実行したり、  
Rubyが処理に困ると、  
エラーメッセージが表示されます。

```
irb(main):001:0> 10/0  
ZeroDivisionError:  
  divided by 0  
    from (irb):2:in `/'  
    from (irb):2
```

日本語訳  
ゼロ除算エラーだよ。  
0で割ってるよ。  
/ のところだよ。

0で割り算するとエラーになるので  
注意してください。

次の話題は  
少しマニアックですが、  
心のどこかに留めて  
おいてください

# Floatオブジェクトは不正確

0.1を3回足したら0.3になりますよねもちろん

```
irb> 0.1+0.1+0.1 == 0.3  
=> false
```

`==` は 等しいかどうか調べる演算子

0.1を3回足したら0.3にならなかった

Rubyの中の0.1は正確に数学上の0.1と一致しない

※整数(Fixnum)オブジェクトは正確です。

小数を正確に計算したい場合は、  
整数にして計算するか、  
**Rational(有理数)オブジェクトを使う**

**Rational(1/10) +**

**Rational(1/10) +**

**Rational(1/10) == Rational(3/10)**

**=> true**

1/10を3回足したら3/10になった

**Rational(分母/分子)**

では、  
irb の世界から抜けて、  
shell の世界へいきます。

# 3つの世界

**Ruby(irb)**

Ruby語が通じる世界  
1行ずつコードを実行

**Ruby(ファイル)**

Ruby語が通じる世界  
ファイルにコードを書いて実行



**Shell**

OS語が通じる世界  
ターミナルの中

# irb の世界と shell の世界

irb の外は shell という世界

\$ irb

shell語が通じる世界 shell

2.0.0p0 :001 > puts 1+2+3

6

=> nil

Ruby語が通じる世界

2.0.0p0 :002 > exit

irb

\$

shell語が通じる世界 shell

# 世界を移動するコマンド

\$ irb

shell語が通じる世界 shell

2.0.0p0 :001 > puts 1+2+3

6

=> nil

2.0.0p0 :002 > exit

Ruby語が通じる世界

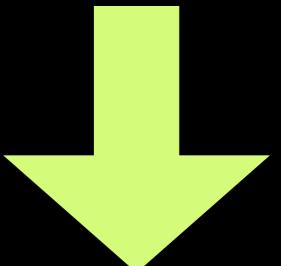
irb

\$

shell語が通じる世界 shell

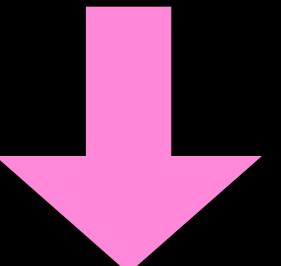
irb

コマンド



exit

コマンド



では、shellについてもう少し詳しく説明します。

**shellはOSと話をするための世界です。**  
(OS=パソコンと考えてもらっても大体あってます)  
**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

**pwd** : 今いるフォルダ名を表示

**cd** : フォルダー移動

**mkdir** : フォルダー作成

# ls: ファイル一覧を見る

```
$ ls
```

ファイル一覧を表示します。フォルダーも見れます。

オプション: ls のあと 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)

# 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。

# cd：フォルダー移動

フォルダー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 : フォルダー作成**

フォルダー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**



**shellの話はひとまずここまで。  
また必要なときに説明します。**  
次はファイルに  
**Rubyコードを書いてみましょう。**

# ファイルにRubyコードを書いて実行

## Ruby(irb)

Ruby語が通じる世界  
1行ずつコードを実行

## Ruby(ファイル)

Ruby語が通じる世界  
ファイルにコードを書いて実行

## Shell

OS語が通じる世界  
ターミナルの中

コード実行時にshellを利用します。

# 0. 前準備 Windows

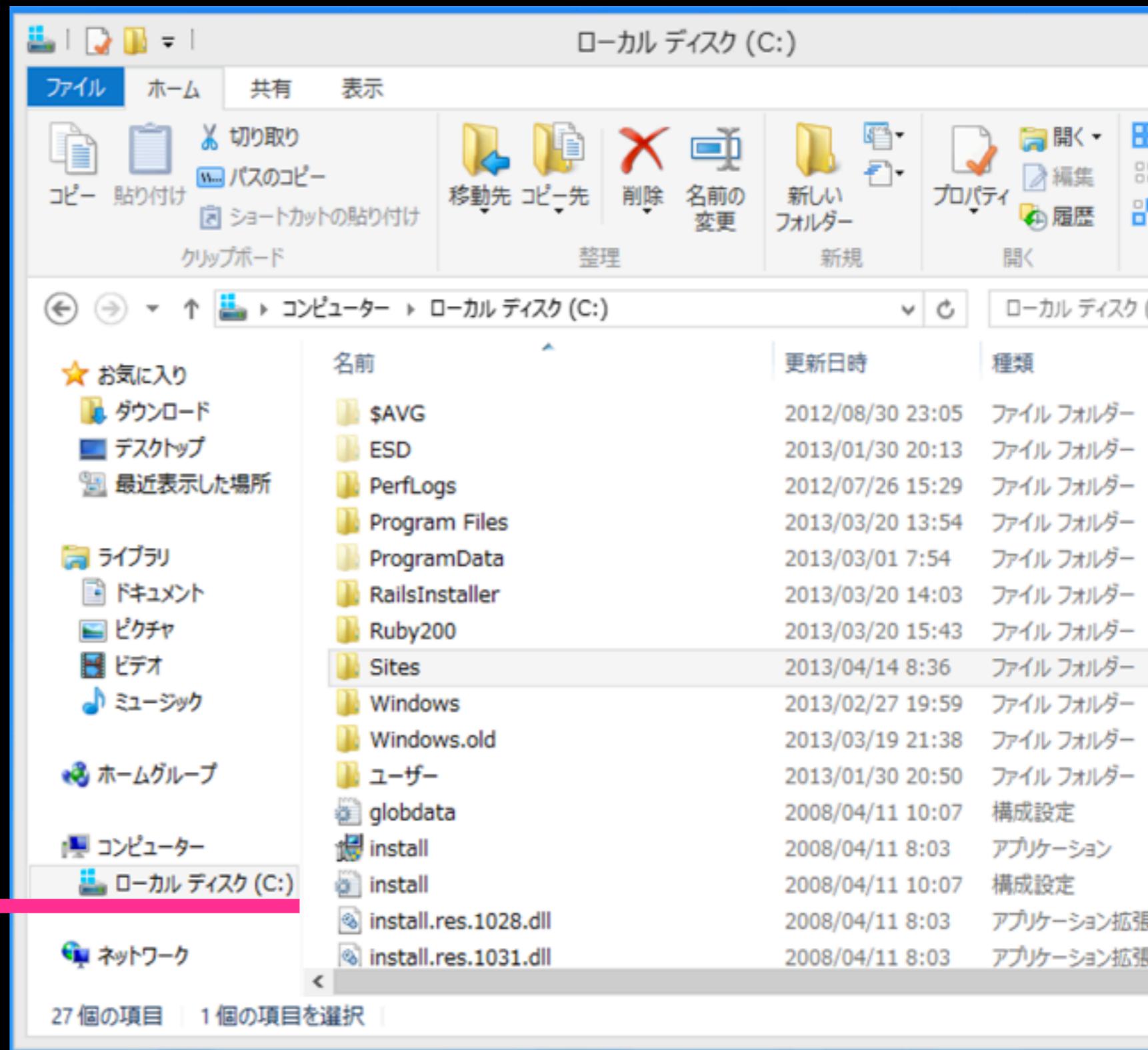
これからのお講義で書くコードを置く場所を作つておきます。

※以下、お勧めの場所を書きますが、詳しい人は自分の好きな場所で構わないです。

C:¥Sitesフォルダ  
の中に好きな名前で  
フォルダを作つてく  
ださい。

(例) C:¥Sites¥src

※ C:¥ フォルダはサ  
イドバーの下の方にあ  
るのでスクロールして  
探してください。



# 0. 前準備 Mac

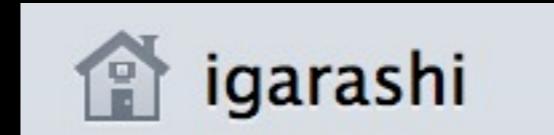
これからのお講義で書くコードを置く場所を作つておきます。

※以下、お勧めの場所を書きますが、詳しい人は自分の好きな場所で構わないです。

Mac : home フォルダの下に好きな名前でフォルダを作つてください。(例) /Users/igarashi/src

※Finderにhomeが出ない場合は以下で設定できます。

Finderを起動して画面上部メニューの Finder - 環境設定 選択



home フォルダ

←home フォルダにチェック

# 0. 前準備 Linux(Ubuntu)

これからのお講義で書くコードを置く場所を作つておきます。

※以下、お勧めの場所を書きますが、詳しい人は自分の好きな場所で構わないです。

Linux(Ubuntu) : home フォルダの下に好きな名前  
でフォルダを作つてください。(貸出PCの人は自分の名  
前をつけるといいかも。) (例) /home/dev/src



# Rubyコードをファイルに記述して実行

## 1. エディタを起動します。

インストールしたエディタを起動します。

お勧めエディタ

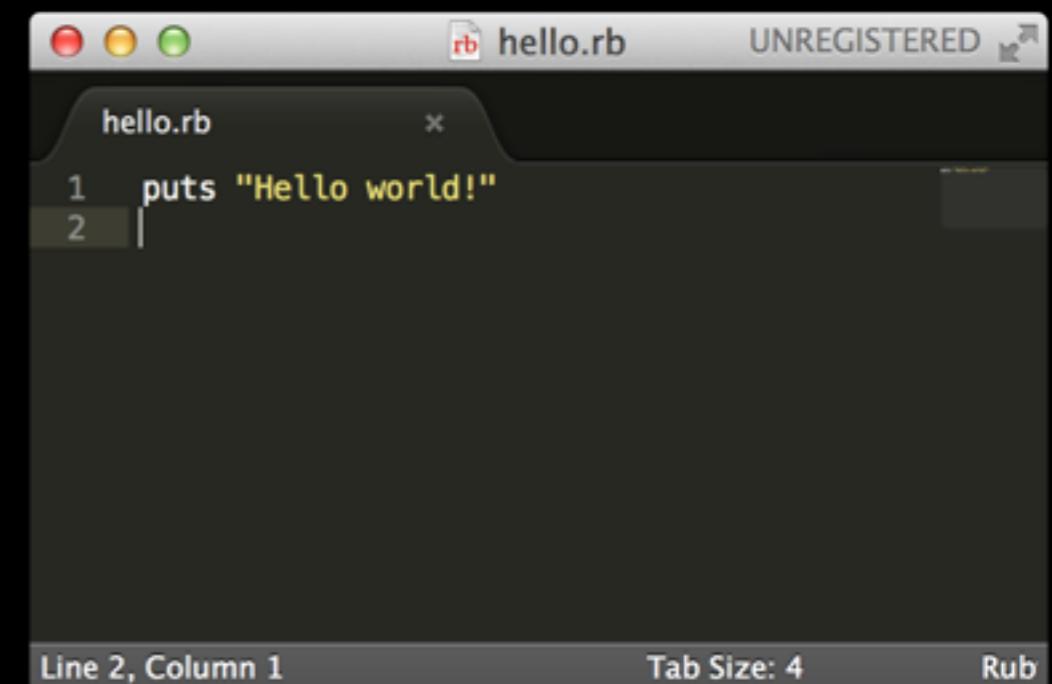
Win, Mac : SublimeText2

Linux(Ubuntu) : gedit

(Ubuntu は次ページに解説あります。)

## 2. プログラムを入力します。

```
puts("Hello world!")
```



A screenshot of the Sublime Text 2 interface. The window title is 'hello.rb'. The code editor contains the following text:

```
1 puts "Hello world!"
```

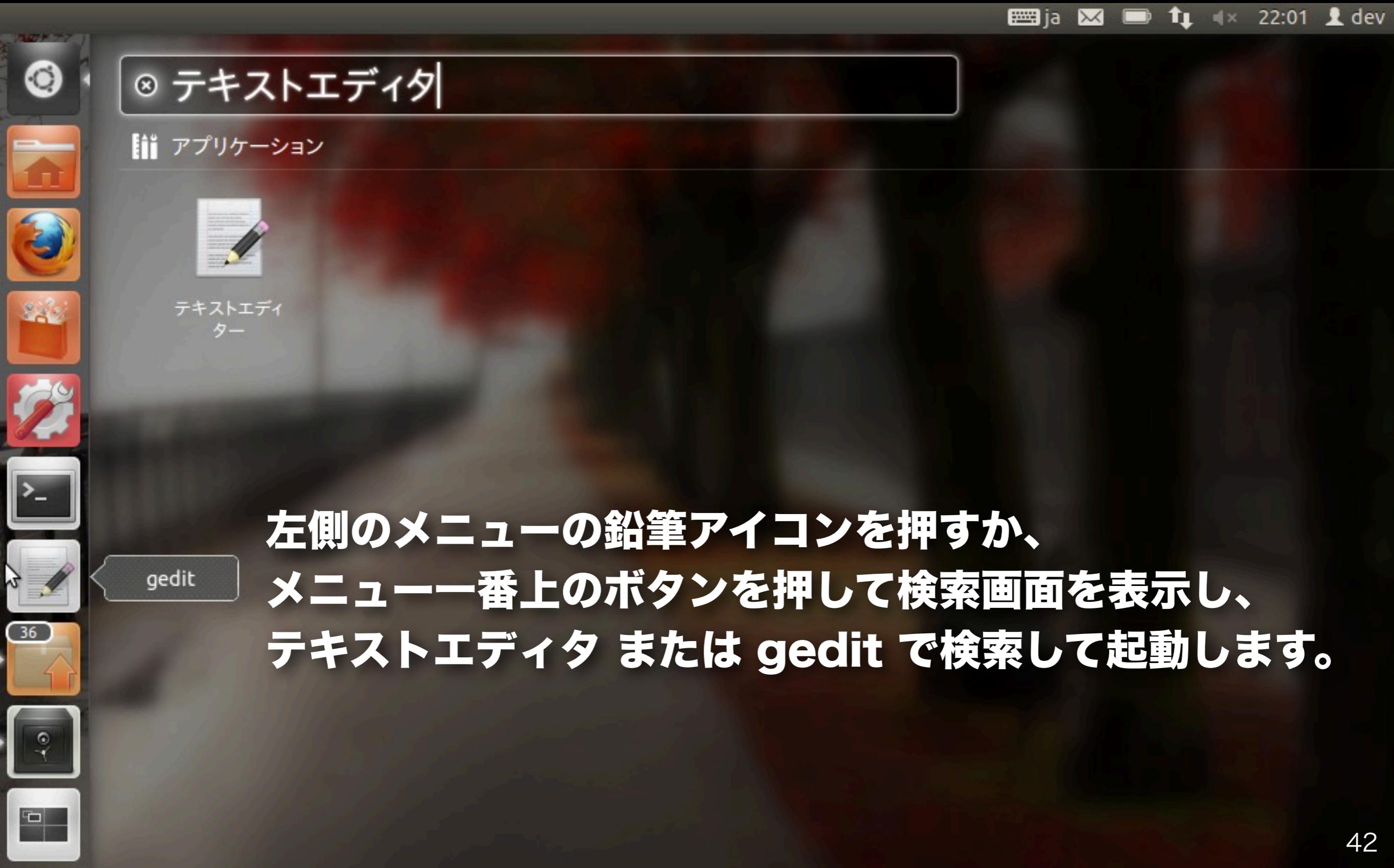
The status bar at the bottom shows 'Line 2, Column 1' and 'Tab Size: 4'. The tab bar at the top right shows 'UNREGISTERED'.

## 3. hello.rbという名前で保存します。

手順0. 作ったフォルダの下に保存してください。

※名前は .rb を付ければ、他の名前でもOKです。

# Linux(Ubuntu) gedit起動



# Rubyコードをファイルに記述して実行

4. ターミナルを起動します。

5. cd コマンドでhello.rb を保存したフォルダへ移動します。

\$ cd src

※pwdコマンドを使うと現在のフォルダを確認できます。 \$ pwd

6. hello.rb を実行します。

\$ ruby hello.rb

※ruby [ファイル名] で実行

Hello world! と表示されれば成功です。

成功例

\$ ruby hello.rb ↵enter  
Hello world!

※以下のエラーが出たときは 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)

# ファイルにRubyコードを書いて実行 Rubyの世界とshellの世界

Ruby語(rubyコード)はファイルに書きます。

hello.rb ファイルの中はRuby語が通じる世界

```
puts "Hello world!"
```

ruby code が書かれたファイルをshellで実行します。

```
$ cd src
```

shell語が通じる世界 shell

```
$ ruby hello.rb
```

# ターミナルを起動した時の始点フォルダ

Windows : /C/Sites (C:¥Sites)

Mac : /Users/ユーザー名/

Linux(Ubuntu) : /home/ユーザー名/

# エラーメッセージ

正しくないプログラムを実行しようとすると、  
エラーメッセージが表示されます。  
以下のコードを実行してみてください。

`helloerror.rb`

```
print("Hello, Ruby.\n")
prin("Hi.") ←正しくない
```

`$ ruby helloerror.rb`

```
Hello, Ruby.
helloerror.rb:2:in `<main>': undefined
method `prin' for main:Object
(NoMethodError)
```

# 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

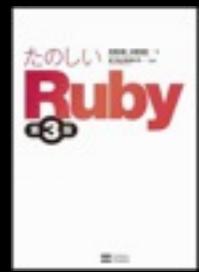
**helloerror.rb:2:in `<main>': undefined  
method `prin' for main:Object  
(NoMethodError)**

日本語訳

**helloerror.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  
**そんなメソッドないよエラー** が起きたよ

# 追加演習

# 日本語と文字コード



教科書  
p.16

Rubyコード中に日本語を書く場合は、1行目に

# encoding: utf-8

などと書く必要があります。(マジックコメントと呼びます)

**chihayafu.rb**

```
# encoding: utf-8
```

```
puts("ちはやふるかみよもきかずたつたがわ")
```

```
puts("からくれないにみづくくるとは")
```

```
$ ruby chihayafu.rb
```

ちはやふるかみよもきかずたつたがわ  
からくれないにみづくくるとは

# 文字コード

コンピュータで日本語を扱う場合には  
いくつかの表現方法があります。  
これを 文字コード と呼びます。

## 文字コードの種類

UTF-8 : 基本はこれでOK。スタンダードな方法。  
CP932(Shift JIS) : Windowsで使われる。

SublimeText2 や gedit ではデフォルト(初期設定)が(Win でも)UTF-8 になっているので、特に変更する必要はないです。  
※なんらかの理由で変更する場合は File メニューの  
save with Encoding を選びます。

# マジックコメント

そのファイルの文字コードをRubyへ教えるために  
1行目に書くおまじないです。

例)

**UTF-8 # encoding: utf-8**  
**Shift JIS # encoding: Shift\_JIS**

```
# encoding: utf-8 ←マジックコメント
puts("ちはやふるかみよもきかずたつたがわ")
puts("からくれないにみづくくるとは")
```

# 今日の講義のまとめ

# 3つの世界

Ruby(irb)

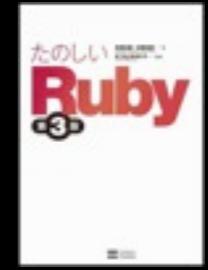
Ruby語が通じる世界  
1行ずつコードを実行

Ruby(ファイル)

Ruby語が通じる世界  
ファイルにコードを書いて実行

Shell

OS語が通じる世界  
ターミナルの中



教科書  
p.8,9

`puts("Hello, World!")`

オブジェクト

`String(文字列)` オブジェクト

# `puts("Hello, World!")`



教科書  
p.8,9

## 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

"Hello, World!" はString(文字列)オブジェクトであり、この場合はメソッドへ渡しているので引数でもあります。

また、引数の前後についているカッコ () は(原則)省略可能で、() の代わりにスペースで書くこともできます。  
つまり、以下の2文は同じ意味です。

```
puts("Hello, World!")
puts "Hello, World!"
```

# irb (Interactive RuByの略)

irb はrubyコードを1行ずつ実行する環境です。

起動方法

Mac : ターミナル

Win : "Command Prompt with Ruby and Rails"

Linux(Ubuntu) : 端末(※分からぬ場合は次のページ参照)

※ ↑ どれも今後はターミナルと呼びます。 (ターミナルの和訳が端末)

起動して以下のように入力してEnterを押してください。

\$ irb

※先頭の \$ はターミナルであることを表すマークです。  
(\$と、その次のスペースは入力不要です。 irbと打てばOK。)  
今後、ターミナルで打つコマンドは同じ書式  
(紫の背景色、 \$ マーク)で書きます。

こんな風に表示が出ればOKです。

2.0.0p0 :001 >

※irbを終了させるには exit と打ちます。



# 四則演算

```
puts 1 + 2
```

```
puts 2 - 3
```

```
puts 5 * 10
```

```
puts 100 / 4
```

```
puts 2**32
```

+ : 足し算

- : 引き算

\* : 掛け算

/ : 割り算

\*\* : 累乗

logとかsinとかもあります。

知りたい方はこちら。

<http://miyamae.github.io/rubydoc-ja/2.0.0/#!/class/-math.html>

**shellはOSと話をするための世界です。**  
(OS=パソコンと考えてもらっても大体あってます)  
**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

**pwd** : 今いるフォルダ名を表示

**cd** : フォルダー移動

**mkdir** : フォルダー作成

# Rubyコードをファイルに記述して実行

## 1. エディタを起動します。

インストールしたエディタを起動します。

お勧めエディタ

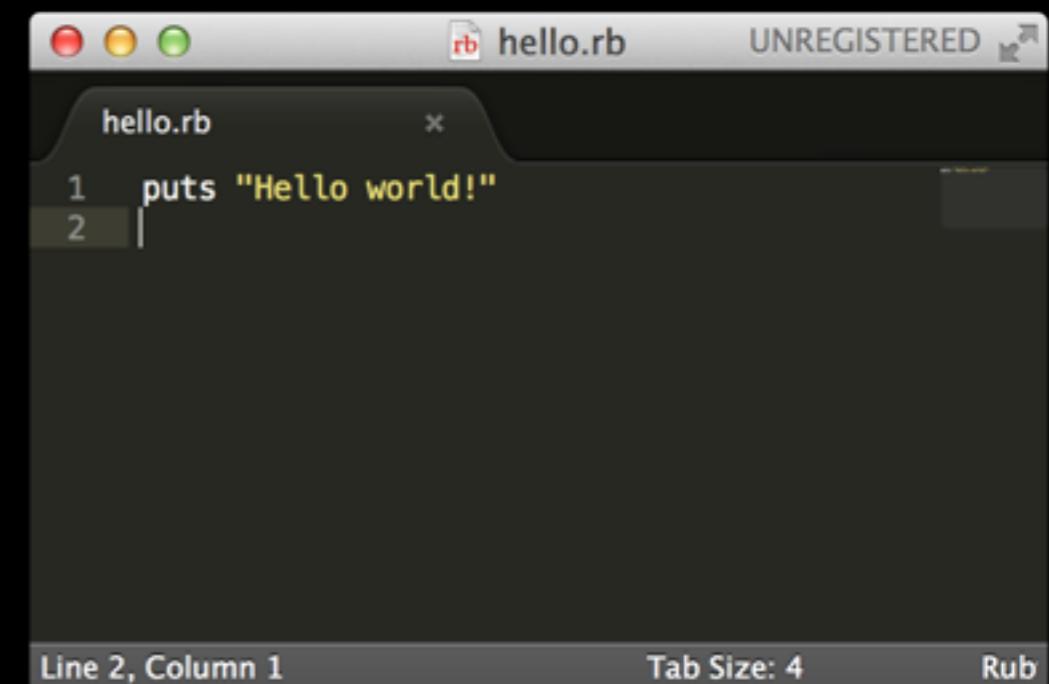
Win, Mac : SublimeText2

Linux(Ubuntu) : gedit

(Ubuntu は次ページに解説あります。)

## 2. プログラムを入力します。

```
puts("Hello world!")
```



A screenshot of the Sublime Text 2 interface. The window title is 'hello.rb'. The code editor contains the following text:

```
1 puts "Hello world!"
```

The status bar at the bottom shows 'Line 2, Column 1' and 'Tab Size: 4'. The word 'Ruby' is also visible in the status bar.

## 3. hello.rbという名前で保存します。

手順0. 作ったフォルダの下に保存してください。

※名前は .rb を付ければ、他の名前でもOKです。

# Rubyコードをファイルに記述して実行

4. ターミナルを起動します。

5. cd コマンドでhello.rb を保存したフォルダへ移動します。

**\$ cd src**

※pwdコマンドを使うと現在のフォルダを確認できます。 **\$ pwd**

6. hello.rb を実行します。

**\$ ruby hello.rb**

※ruby [ファイル名] で実行

Hello world! と表示されれば成功です。

成功例

\$ ruby hello.rb ↵enter  
Hello world!

※以下のエラーが出たときは 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)**

# 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

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

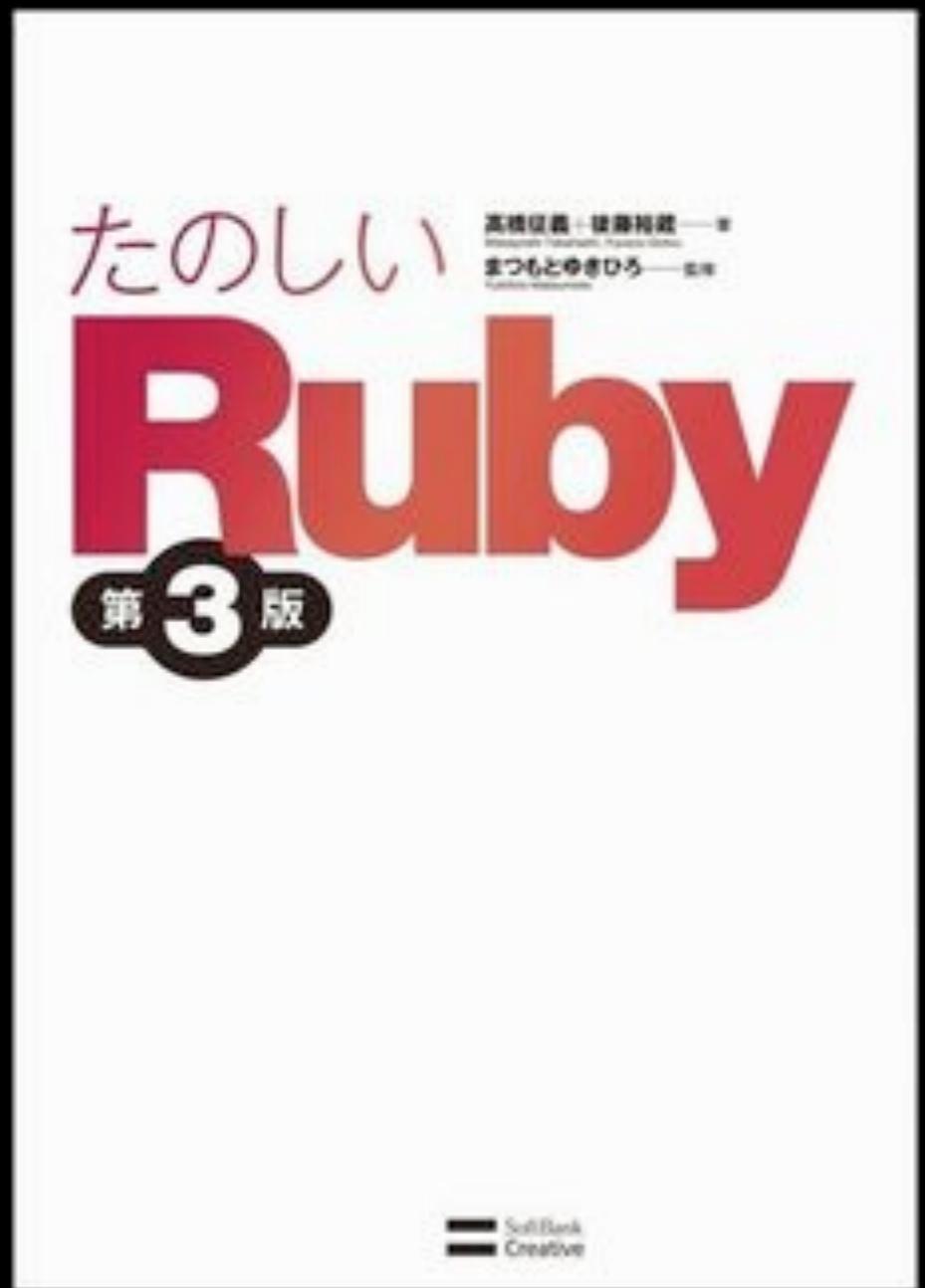
日本語訳

**helloerror.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  
**そんなメソッドないよエラー** が起きたよ



# 參考資料

# 教科書：たのしいRuby



<http://www.amazon.co.jp/dp/4797357401/>



お買い求めは  
大学生協または  
ジュンク堂池袋店で

# 講義資料置き場

過去の資料がDLできます。

<https://github.com/igaiga/hitotsubashi-ruby-2013>

# 雑談・質問用facebookグループ

<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
- 参加者はスタッフ(講師・TA)と昨年、今年の受講者です
- 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