

Ruby on Rails 講義

第17回 Rails基礎

Kuniaki IGARASHI/igaiga

2012.10.11 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IV
(ニフティ株式会社寄附講義)

○ 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」の提供を開始

○ 「@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

○ 「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.

ニフティ株式会社

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19日 IR [「@nifty EMOBILE LTE 定額にねんプラン」の提供を開始](#)
- 2012年4月19日 IR [ニフティとサンリオウェーブ、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](#)

提供

五十嵐邦明

講師 株式会社万葉 エンジニア



評価方法

演習課題を提出していただき成績評価を行います。
また、授業への貢献度も評価の対象とします。
出席が授業回数の3分の2に満たない場合は「F」とします。

雑談・質問用facebookグループ

後期もそのまま使えます。思い出したときに雑談や質問してもらってOK。
邪魔なようならグループから退会してください。

<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
- 雜談、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

講義資料置き場

過去の資料がDLできます。

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

or

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

自習用Rails資料

後期は特に教科書は使いませんが、自習用には以下の資料をお勧めします。

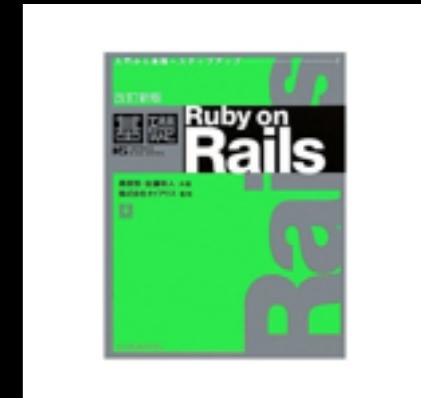
- ▶ **Railsbridge (web)**
- ▶ <http://curriculum.railsbridge.org/curriculum/curriculum>

- ▶ **Rails Guide (web)**
- ▶ <http://guides.rubyonrails.org/>

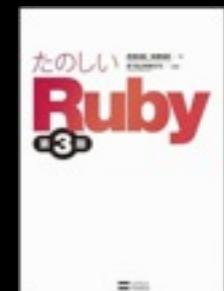
- ▶ **RailsによるアジャイルWebアプリケーション開発 第4版**
- ▶ <http://www.amazon.co.jp/dp/4274068668/>



- ▶ **改訂新版 基礎Ruby on Rails**
- ▶ <http://www.amazon.co.jp/dp/4844331566/>



- ▶ **たのしいRuby**
- ▶ <http://www.amazon.co.jp/dp/4797357401/>



宣传

書いた本が先日 出版されました

- chapter 1 「たのしい開発」を求めて
- chapter 2 Ruby の基礎知識
- chapter 3 Ruby を使ってみよう
- chapter 4 Ruby on Rails とは
- chapter 5 Rails を触ってみよう
- chapter 6 Ruby の文化
- chapter 7 自動化されたテスト
- chapter 8 アジャイル開発と Ruby
- chapter 9 Ruby のコミュニティ
- chapter 10 ある企業の Ruby 導入事例
- chapter 11 「たのしい開発」の答え

たのしい開発 スタートアップ

大場寧子 監修
櫻井達生
大場光一郎
五十嵐邦明 著

これから Ruby はもっと楽しくなる。

たのしい開発

スタートアップ

Ruby

大場寧子 監修
櫻井達生
大場光一郎
五十嵐邦明 著

これから Ruby を始める人、必読。



Ruby の良さがピント來ない人、必読。

Ruby をもっと使いこなしたい人、必読。

まつもと
ゆきひろ氏
高橋征義氏
推薦！





まつもとゆきひろさんご推薦!

Contents

- chapter 1 「たのしい開発」を求めて
- chapter 2 Ruby の基礎知識
- chapter 3 Ruby を使ってみよう
- chapter 4 Ruby on Rails とは
- chapter 5 Rails を触ってみよう
- chapter 6 Ruby の文化
- chapter 7 自動化されたテスト
- chapter 8 アジャイル開発と Ruby
- chapter 9 Ruby のコミュニティ
- chapter10 とある企業の Ruby 導入事例
- chapter11 「たのしい開発」の答え



大場寧子 監修
櫻井達生 著
五十嵐邦明 著

大場
櫻井
大場
五一



Contents

- chapter 1 「たのしい開発」を求めて
- chapter 2 Ruby の基礎知識
- chapter 3 Ruby を使ってみよう
- chapter 4 Ruby on Rails とは
- chapter 5 Rails を触ってみよう
- chapter 6 Ruby の文化
- chapter 7 自動化されたテスト
- chapter 8 アジャイル開発と Ruby
- chapter 9 Ruby のコミュニティ
- chapter 10 とある企業の Ruby 導入事例
- chapter 11 「たのしい開発」の答え

プログラマになる前の筆者は、ソフトウェア開発はとてもたのしいものだと思っていました。新しい技術に触れて、自分のできることが少しずつ増えていき、それまでできなかつたことができるようになること、自分の作ったソフトウェアを使ってもらい、多くの人に喜んでもらうこと、世の中を便利にすること。これらは、とてもワクワクすることなのではないか、そう思っていました。

そんな「開発」にあこがれて、筆者はプログラマになることを決めました。開発の現場には、たのしく効率良く働いて世の中を便利にする人たちがたくさんいるに違いない。そういう人たちとの出会いに胸を膨らませて駆け出しのプログラマとなったのです。

(chapter 1 より)

定価 2,604円

→一橋学生特価

たのしい開発
スタートアップ
Ruby

たのしい開発
スタートアップ
Ruby

大場寧子
橋井達生
大場光一郎
五十嵐邦明 著

大場寧子
橋井達生
大場光一郎
五十嵐邦明 著

たのしい開発
スタートアップ

Ruby



これから Ruby はもっと楽しくなる。

技術評論社

技術評論社

プログラミングは非常に「ハイコンテキスト」な活動です。Ruby によるプログラミングを楽しむためには、実にさまざまな背景知識が必要です。本書では、Ruby プログラミングをより効果的にするための、知識や背景を説明しています。Ruby そのものを「たのしく」解説する書籍は過去にもありました、文化や背景にまで踏み込んだ書籍は本書が最初ではないかと思います。

(まつもとゆきひろ氏「スタートアップRuby」序文より)



500円(81%off!!)

購入ご希望の方は五十嵐までご連絡ください。(予約制)

環境構築

以下の環境構築を済ませておいてください。

Windows, Mac

VM Ubuntu または Rails Installer

(最近、Mac用のRails Installerも出ました。)

どちらを使ってもかまわないです。

こちらでの動作確認はVM Ubuntuを使って
行うので、VMの方がトラブル少ないかも。

Ruby on Rails開発環境

以下のバージョンを想定していますが、講義期間中にバージョンアップするので新しいものに変更になるかもしれません。

- ▶ Ruby 1.9.3 p194
- ▶ Ruby on Rails 3.2.x

(2012年9月29日現在の最新は3.2.8)

VM環境構築

1. USBメモリから .vhd ファイルコピー
or 講義資料置き場からDL

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

2. VirtualBoxをサイトからDL

<https://www.virtualbox.org/wiki/Downloads>

3. 下記のページを参考に起動

<http://www.ubuntulinux.jp/download/ja-remix-vhd>

次ページ以降にポイントを貼ってあります。

VM OS ログインID

id: dev password: devdev



VirtualBox

Download VirtualBox

Here, you will find links to VirtualBox binaries and its source code.

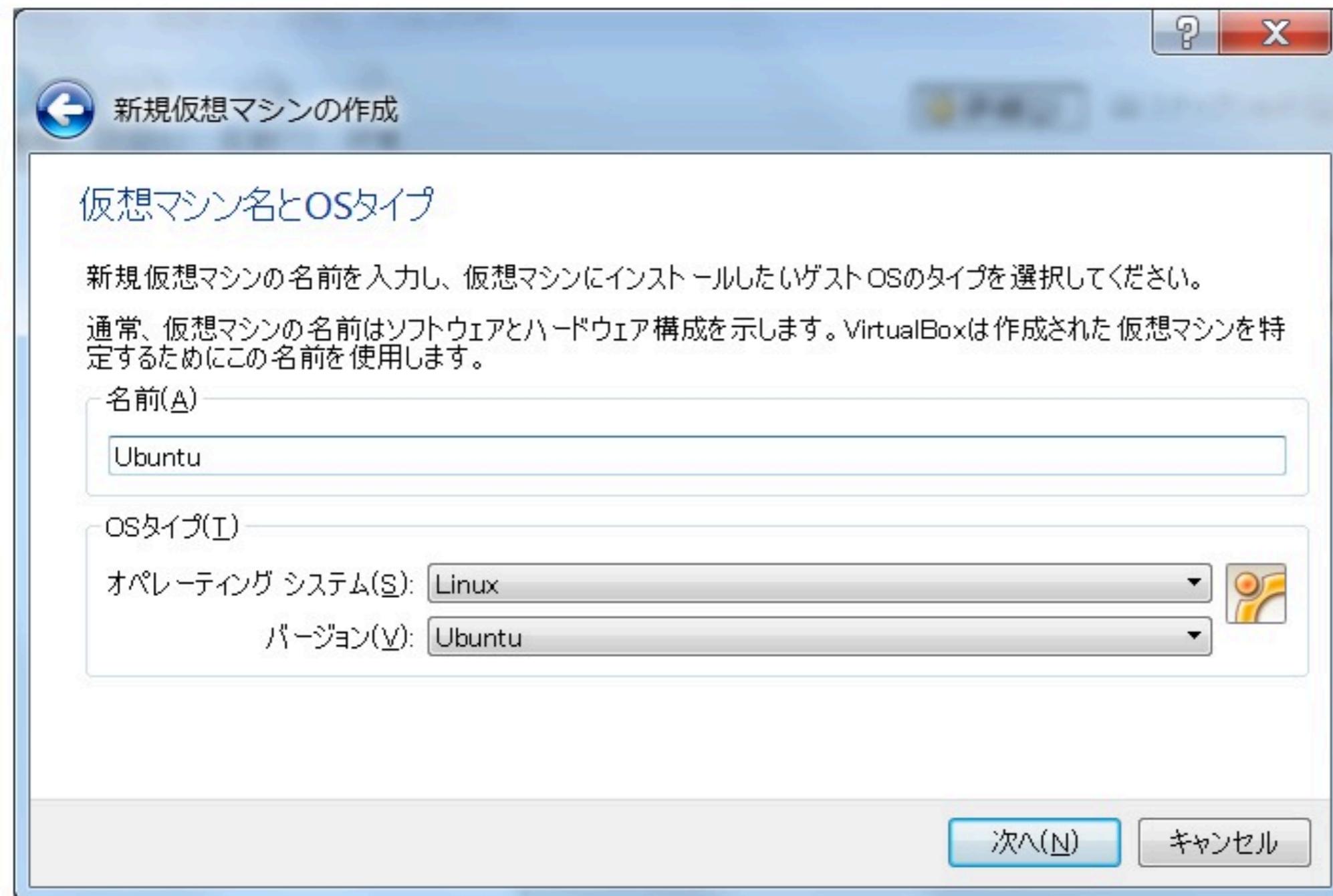
- [About](#)
 - [Screenshots](#)
 - [Downloads](#)
 - [Documentation](#)
 - [End-user docs](#)
 - [Technical docs](#)
 - [Contribute](#)
 - [Community](#)
- ## VirtualBox binaries
- By downloading, you agree to the terms and conditions of the respective binary's license.
- VirtualBox platform packages.** The binaries are released under the Apache License, Version 2.0.
 - [VirtualBox 4.1.22 for Windows hosts](#) ↗x86/amd64
 - [VirtualBox 4.1.22 for OS X hosts](#) ↗x86/amd64
 - [VirtualBox 4.1.22 for Linux hosts](#)
 - [VirtualBox 4.1.22 for Solaris hosts](#) ↗x86/amd64
 - VirtualBox 4.1.22 Oracle VM VirtualBox Extension Pack** ↗All
 - Support for USB 2.0 devices, VirtualBox RDP and PXE boot for Intel PRO/100 MT and PRO/100 MT2 network cards is included in this Extension Pack. The Extension Pack binaries are released under the Apache License, Version 2.0.
 - Please install the extension pack with the same version as your installed VirtualBox host.*
 - If you are using [VirtualBox 4.0.16](#), please download the extension pack for that version.
 - VirtualBox 4.1.22 Software Developer Kit (SDK)** ↗All platforms
 - The SDK includes the VirtualBox API headers, documentation, and developer tools.

<https://www.virtualbox.org/wiki/Downloads>

See the [changes log](#) for what has changed.

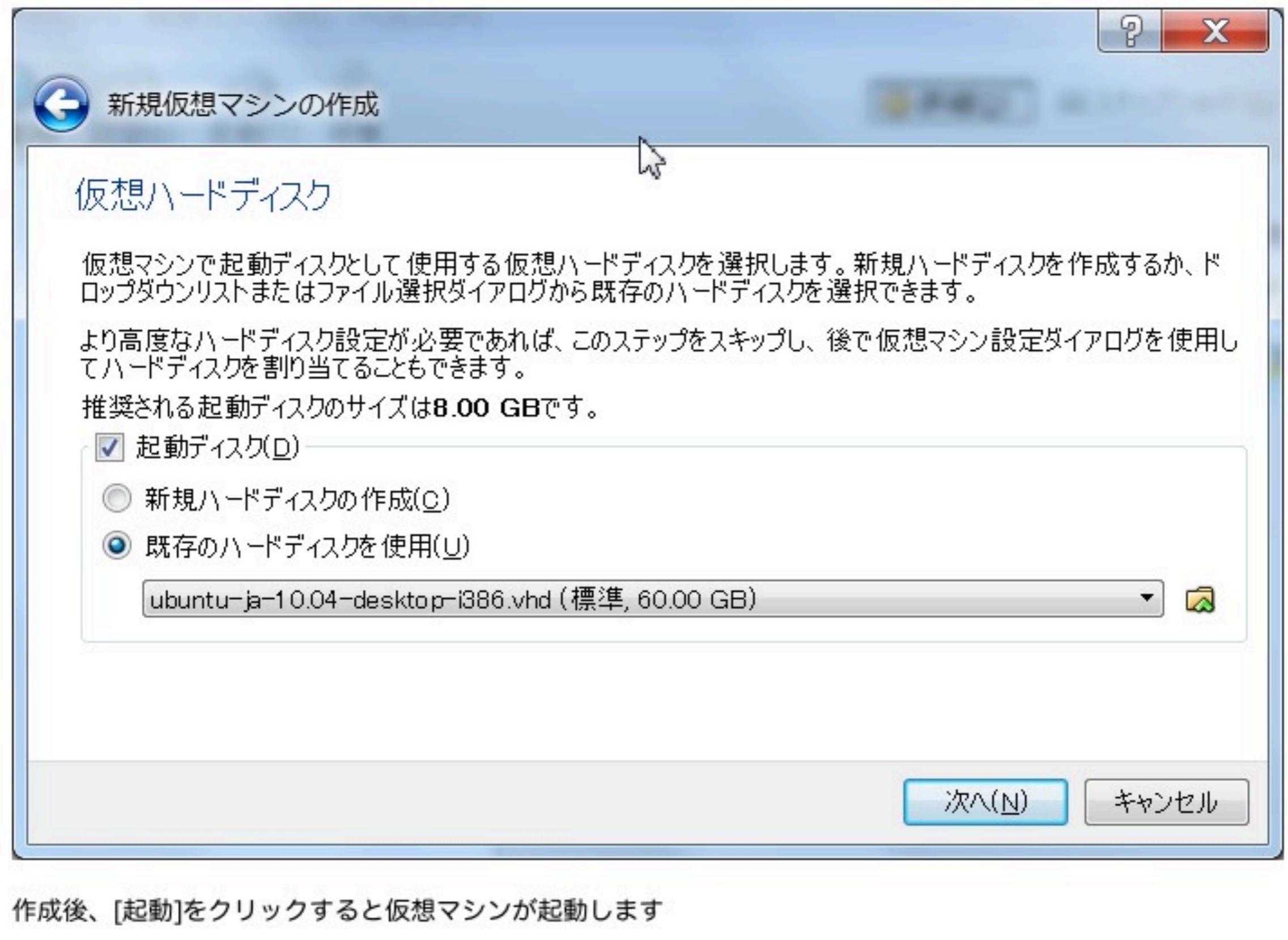
仮想マシンの登録と起動

VirtualBoxを起動し、[新規]をクリックします。仮想マシン作成ウィザードが開くので、画面の指示に従って入力していきます。OSタイプは「Ubuntu」を選んで下さい。



<http://www.ubuntulinux.jp/download/ja-remix-vhd>

[仮想ハードディスク]ダイアログでは、展開した仮想マシンイメージを選びます。



作成後、[起動]をクリックすると仮想マシンが起動します

<http://www.ubuntulinux.jp/download/ja-remix-vhd>

Rails Installer

RAILSINSTALLER GET UP & RUNNING WITH R

STEP 1 

**DOWNLOAD
the KIT**

For OSX.

RailsInstaller has everything you need to hit the ground running. In one easy-to-use installer, you get all the common packages needed for a full Rails stack. Download it now and be writing (and running) Rails code in no time.

- [Ruby](#) 1.9.3-p194
- [Rails](#) 3.2.7
- [Git](#) 1.7.9.6
- [Sqlite](#) 3070500
- [osx-gcc-installer](#) 4.1
- [JewelryBox](#) 1.3.0

STEP 2 

**WATCH
the VIDEO**



STEP 3 

**WHAT'S
NEXT?**

Have troubles installing?
[Check the RailsInstaller Google Group](#)

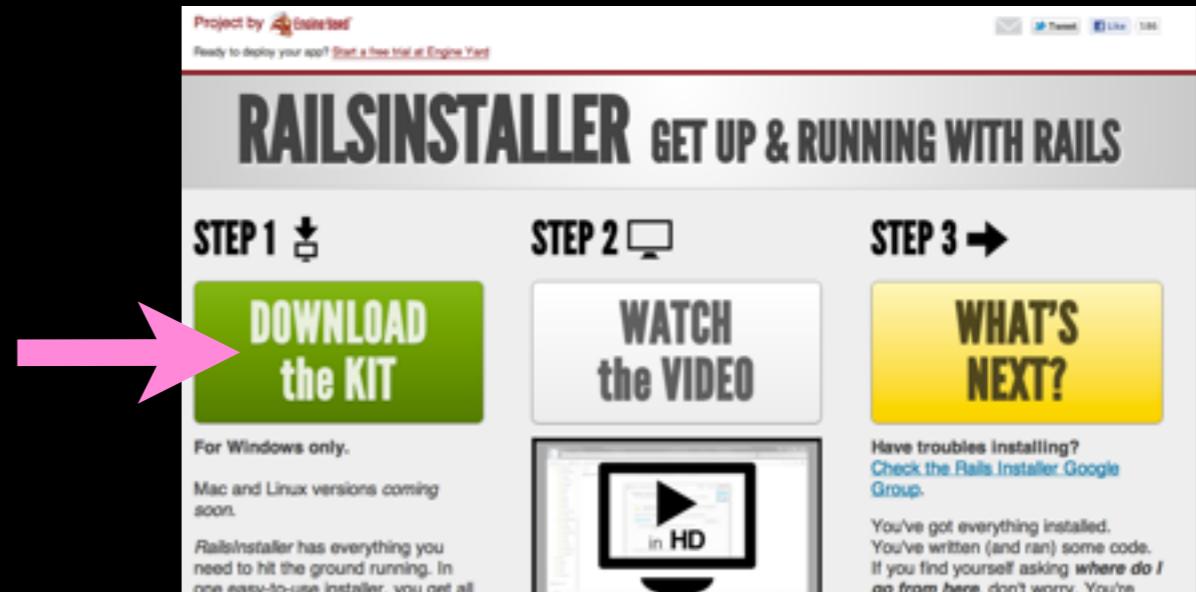
You've got everything installed and written (and ran) some code. If you're still asking **where do I go** from here, don't worry. You're not alone.

The answer is **just about anywhere**. Here's a handful of good places to start:
the official [Ruby on Rails site](#), [blog](#), [account \(@rails\)](#), and [mailing list](#). Drop into IRC, swing by [#rubyonrails](#) on freenode.net. There's always the growing [Ruby on Rails](#) [sites](#).

<http://railsinstaller.org/>

RailsInstaller

1. Railsインストーラーを
Download、Install
<http://railsinstaller.org/>



2. 名前とemailを入力 (公開されます。)
githubやアプリ公開時に使われます。公開されたくない人はニックネームなどを使ってください。

3. 起動(Windows)
Windowsメニューから以下を実行

「プログラムとファイルの検索」から "Command Prompt with Ruby and Rails" or
スタート→全てのプログラム→RailsInstaller→"Command Prompt with Ruby and Rails"

起動(Mac)

アプリケーション - ユーティリティ - ターミナル.app を起動

エディタ - コードを打つ道具

推奨エディタ(お気に入りがある人はそれでOKです)

Windows :

komodo-edit : <http://www.activestate.com/komodo-edit>

intype : <http://inotai.com/intype/>

Ubuntu :

gedit テキストエディター (インストール済)

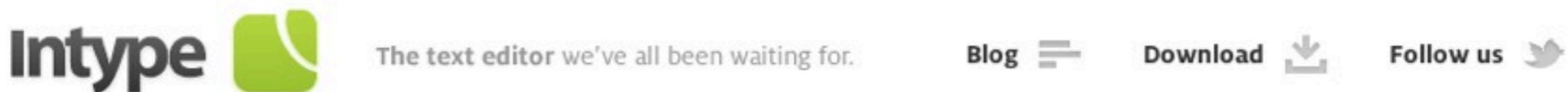
Mac : テキストエディット (インストール済)

komodo-edit : <http://www.activestate.com/komodo-edit>

CotEditor : <http://sourceforge.jp/projects/coteditor/>

intype (Win)

<http://inotai.com/intype/>



**Highly addictive
text editing!**

USE WITH CAUTION

It may cause awesomeness
of your work!

It may cause you to be
overproductive!

It may cause you to hate all
other editors!

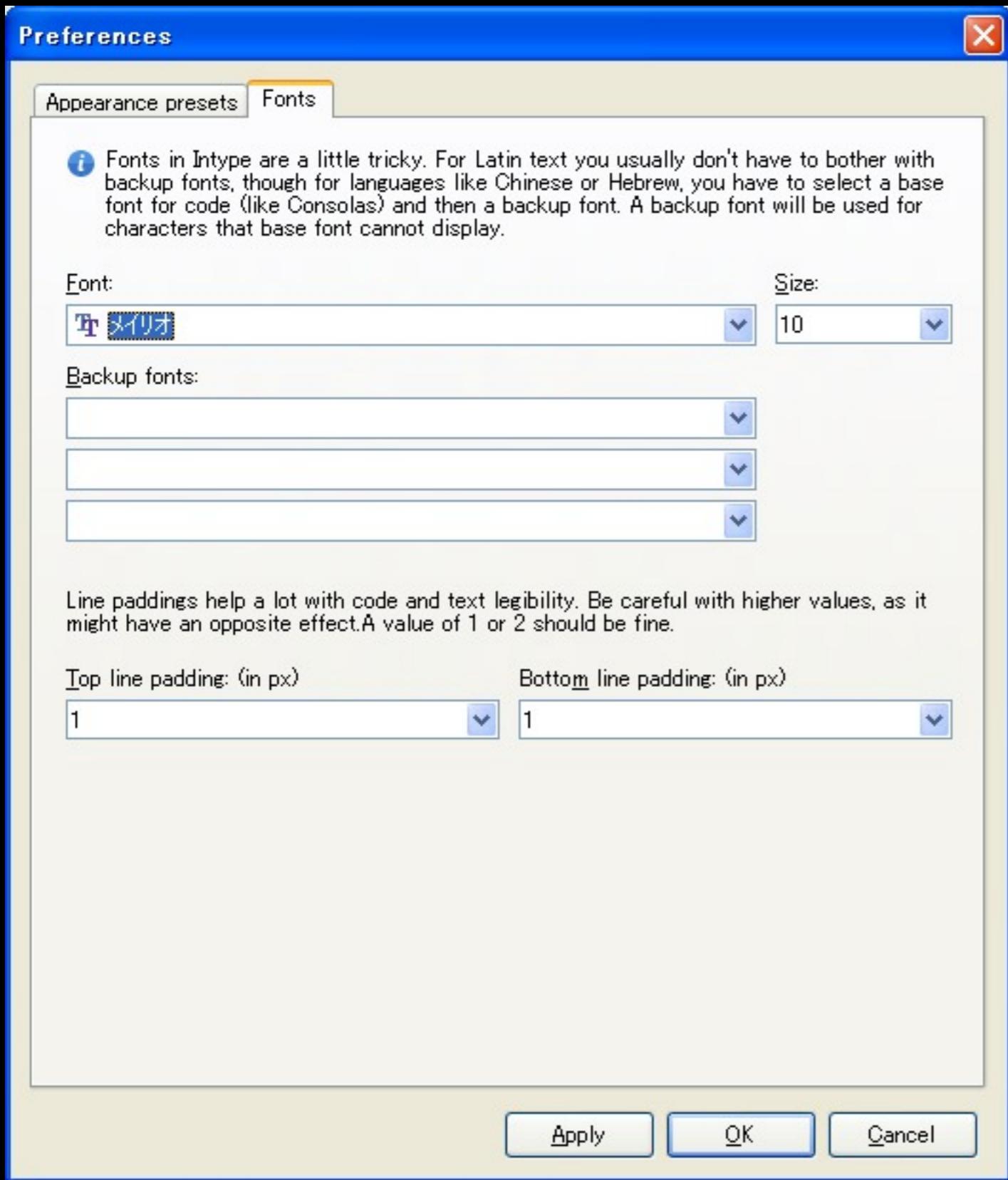
Download now

A screenshot of the Intype text editor interface. The window title is "index.html - D:\work\projects\intype\web\design\2.0\screenshots\projects\Blog\...". The menu bar includes File, Edit, Search, Project, View, Text, Bundles, and Help. The toolbar has tabs for index.html, buttons.css, layout.css, and app.js. The left sidebar shows a project structure for "Blog" with "templates", "css" (buttons.css, forms.css, layout.css, reset.css), "img", "js" (app.js, dojo.js), and "test" (gitignore, htaccess, crossdomain.xml, favicon.ico, humans.txt, index.html). The main editor area displays the HTML code for "index.html":

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5
6   <title>Blog</title>
7   <meta name="description" content="">
8   <meta name="author" content="">
9
10  <link rel="stylesheet" href="css/style.css">
11 </head>
12
13 <body>
14   <div id="container">
15     <header>
16       <!-- Todo -->
17     </header>
18     <menu>
19       <!-- Todo -->
```

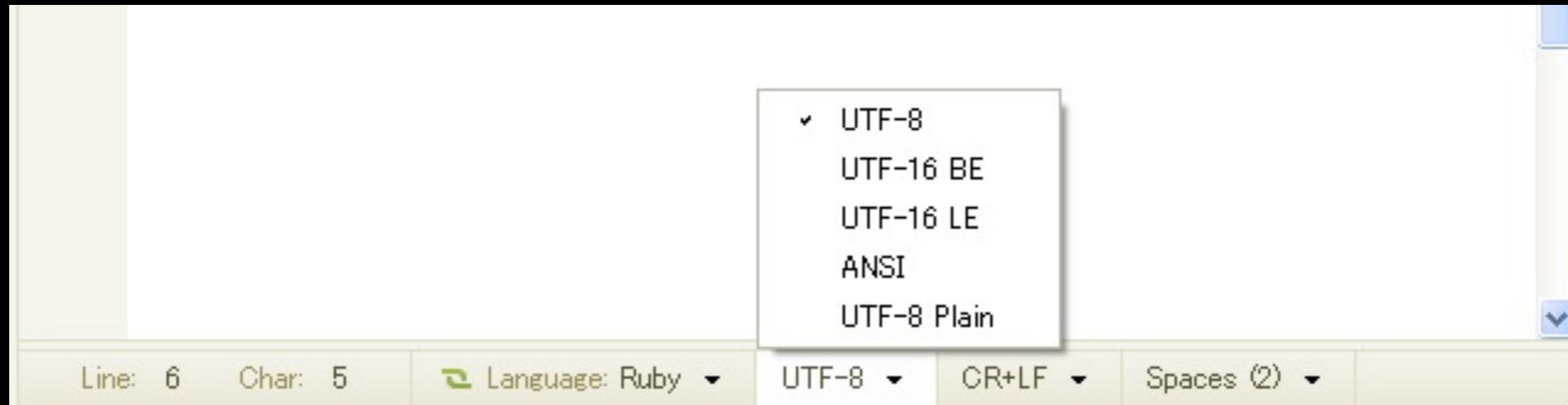
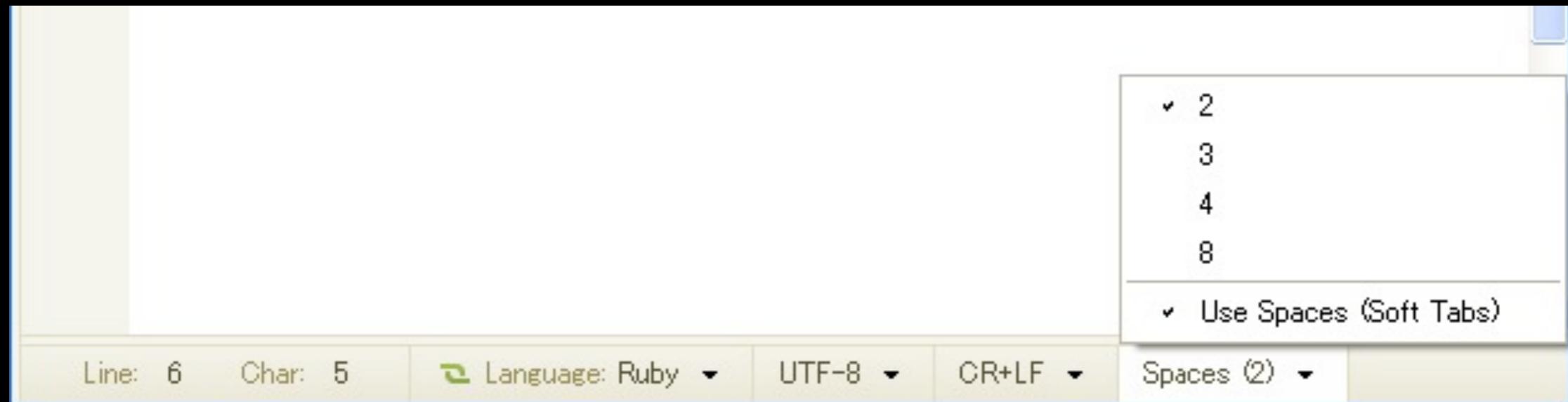
The status bar at the bottom shows "Line: 6 Char: 14" and "Grammar: HTML 5 - ANSI -".

intype



コードを書くための設定
Edit - Preferences
Fonts タブ
Fontを日本語表示できる
ものに変更 (メイリオや
MSゴシック)

intype



Komodo Edit(Win, Mac)

<http://www.activestate.com/komodo-edit>



Code to Cloud: Smarter, Safer, Faster.™

Welcome, Guest Sign In Cart



Stackato Komodo Perl Python Tcl Support & Resources About Us 1.866.631.4581

- What's New >
- Features >
- Komodo Edit**
- Compare Editions >
- Komodo For Teams >
- System Requirements >



ActiveState Komodo® Edit 7

The open source editor for Python, PHP, Ruby, Javascript, Perl and Web development.

Need Help? Email Sales

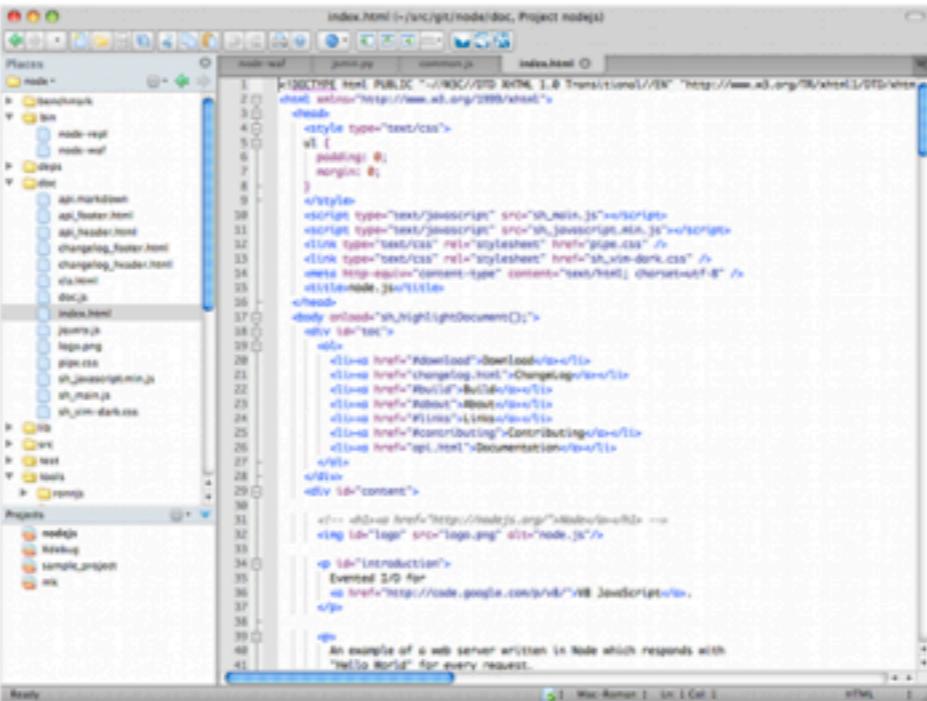
[Download Komodo Edit](#)

[Try Komodo IDE 7](#)

[Buy Komodo IDE 7](#)

[Home](#) > [Komodo](#) > Komodo Edit

Komodo Edit for Python, PHP, Ruby, JavaScript, Perl and Web Dev



Free Code Editor for Windows, Mac and Linux

Komodo Edit is a fast, smart, free and open-source code editor. Switching your trusty code editor is hard, but give Komodo Edit (or its big brother Komodo IDE) a try: it'll be worth your while.

Windows, Mac, Linux? Yes. PHP, Python, Ruby, JavaScript, Perl, Tcl, XML, HTML 5, CSS 3? Yes, with (customizable) syntax coloring, folding, background syntax checking, and intelligent auto-complete and calltips. What else? *Fast Open* (no more slow poking around for files); remote file editing; Vi keybindings (good ones); and a *Toolbox* with shell command integration, macros and code snippets--all wrapped around a sweet editor and Firefox-style extensibility.

[See more of Komodo's features »](#)

目次

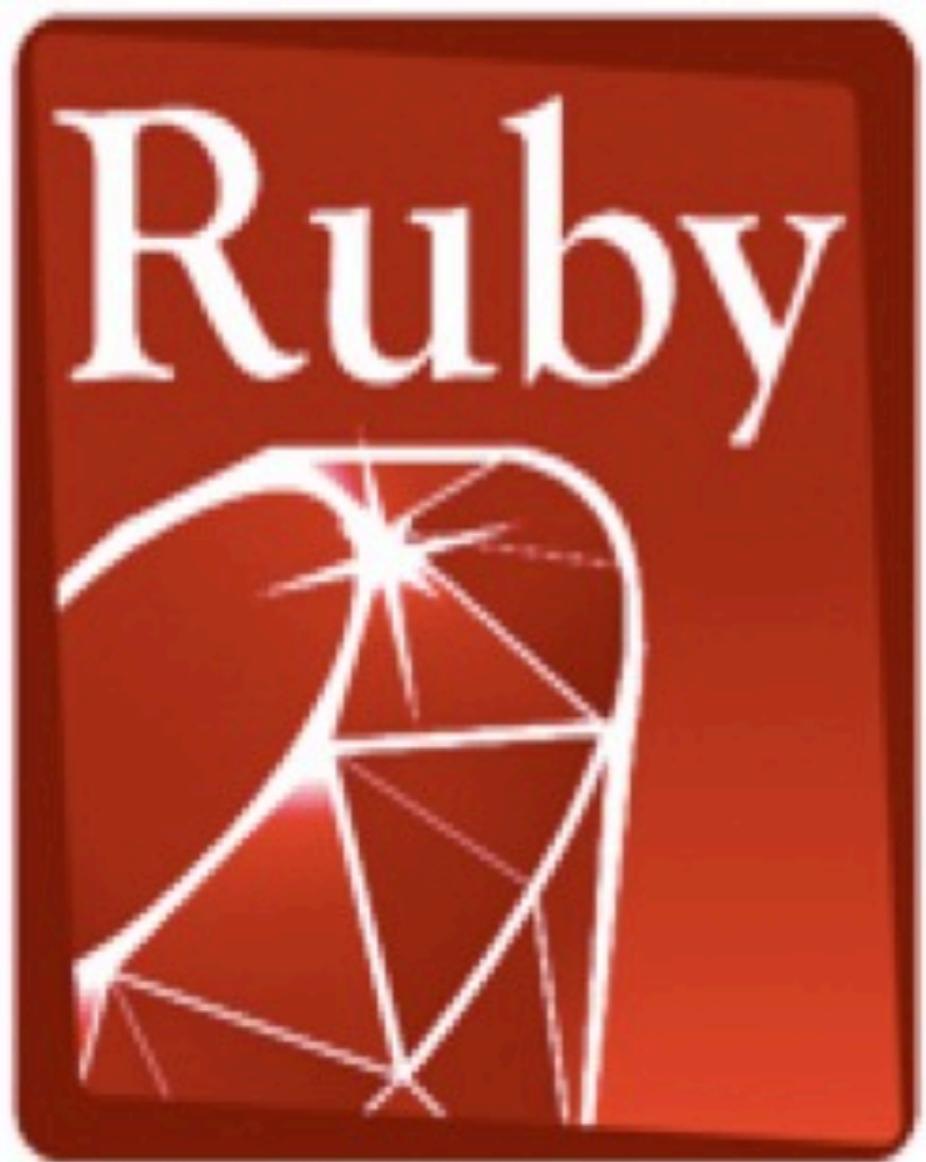
Ruby on Rails 基礎

いちばん小さなRailsアプリ作り

Rubyのコード実行結果をHTMLへ埋め込む

Controller と View

Ruby on Rails とは



Ruby で書かれたWebアプリフレームワーク

Ruby on Railsとは

Ruby で書かれた
Webアプリケーションフレームワーク
= Webアプリケーションを
簡単に作れる仕組み

Webアプリって？

ブラウザでアクセス
できるアプリ



餅 せんざい 七草粥 おしるこ 黒豆

[検索](#)
[人気検索](#)

 クックパッドへ
ようこそ

 みんなのオススメ
レシピ

 あなたのレシピを
のせる

[クックパッドID \(無料\) を登録する](#) | [ログイン](#)

毎日の料理を楽しみに クックパッド

今日はパパの休肝日！出汁にこだわったお味噌汁をご用意♪



本日のピックアップレシピ

[残ったおせち活用☆黒豆きなこシフォン](#) by ヒメとそちゃん

黒豆の和風な甘さときなこの香りたっぷりのしっとりシフォンです☆...

[土 七草粥で今年も健康な一...](#) by momorecipe

[金 *ダシが決め手！うちの...](#) by shokenママ

[木 和風豆腐ハンバーグ*~...](#) by *夏海*

[水 鶏肉と野菜のポン酢煮込...](#) by rinocojp

[火 かぶのほっこり*和風力...](#) by patisier

[月 母が作る*父のふるさと...](#) by kaopomme♪

話題のレシピ

つくれば10人おめでとう！

・[豚バラかたまり☆簡単！...](#) 20:38

by kana89626

・[フライドポテト風餃...](#) 20:21

by ぢる助

・[*可愛いマイメロちゃん...](#) 19:43

by ブラック☆ラビ

・[簡単♪ヘルシー♪お豆腐...](#) 19:37

by ぶりぶりぶりん

・[【キャラ弁*可愛いハー...](#) 19:04

by ブラック☆ラビ

・[豆腐と人参の卵炒め \(簡...](#) 18:37

by kuro羊

・[あられ大根と梅の納豆餅](#) 18:23

by カルン

[バックナンバー»](#)

受賞レシピが

はなまる×クリッキー
料理選手権



我が家家のNo.1おかず

この広告を評価してください

イーモバイルが月3,217円で

@niftyの最大42M高速ポケット
1/31迄！今だけ12,000円キャッシュバック

www.nifty.com

"買取王子"楽チン買取

電話一本で自宅まで集荷にお伺い
で色々なものを宅配買取

www.kaitoriouji.jp



お夜食にぴったり♪



手軽でかわいい！



肉と野菜をバランスよく



あっという間に一品完成



お店級♪



あつあつ♪



レンジ♪



特集！

 大根
レシピ

COOKPAD

ただの片づけ術ではあります
モノへの執着を捨て、心もスッキリ
され
Yamada

Rails製

自宅でメガネを試してから買う 5本まで無料でお試しできます

くわしく

閉じる

知的に働く
メガネこの5本を
まとめて試
すヴィヴィッドムーン VM-
11356-100
25,000円ブルガリ BV-1034T-
4047
52,500円コーチ FRANCESCA
30,450円ディオール CD3759-
PDE
26,000円アミバリ TR372-14
27,300円デキるビジ
ネスマンこの5本を
まとめて試
す元 GEN-50-10
33,000円美 SHAPE SN-002-1
23,000円ジーパーツ Z-35
23,100円元 GEN-50-14
33,000円美 SHAPE SN-002-2
23,000円個性派クリ
エイティブこの5本を
まとめて試
すアランミクリ AL1053-
0202
42,000円トラクション GEMINI
Tourmaline
32,550円アクセル RIVA235
34,650円プラトイ TeA-2
31,500円59 ヒステリック
CHAPLIN-gold
21,000円凛々フェミ
ニンこの5本を
まとめて試
すブルガリ BV-1034T-
4029
52,500円ティファニー TF-46A-
8124
31,650円シモンズ S-103-
12125円ルイ・ヴィトン LV-041-103
26,750円Rails製
26,250円

Oh my Glasses

 ユーザー名、またはメールアドレス パスワード 保存する・パスワードを忘れた場合はこちら

Twitter始めませんか？登録する

 名前 メールアドレス パスワード

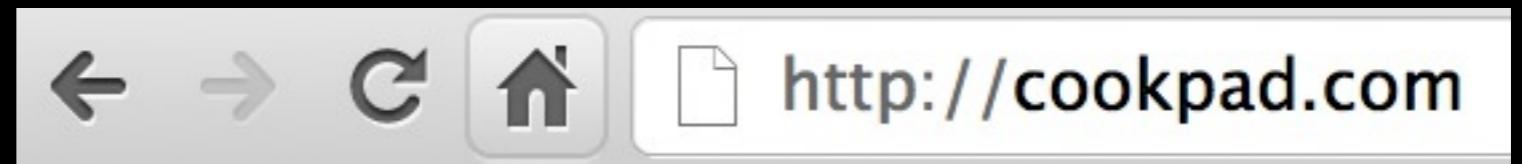
twitter 元Rails製？

では、夏学期の最後の方でやったことの復習からスタートします。

Webアプリの仕組みの 簡単な説明

Webアプリの基本動作

ブラウザにURLを
入力してEnter



リクエスト

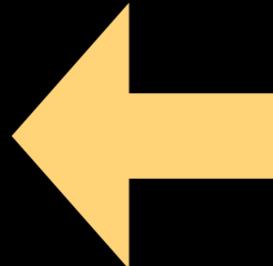
URL : http://cookpad.com/

Web Server

Web App

Browser

http://cookpad.com/



リクエストが飛びます。

Webアプリの基本動作

レスポンスとしてHTMLが返ってくる

リクエスト

URL : <http://cookpad.com/>

Web Server

Web App

レスポンス
HTML

Browser

<http://cookpad.com/>



HTML

HyperText Markup Language

Webページを記述するための言語

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2
3 <html>
4 <head>
5 <script type="text/javascript">var __rendering_time = (new Date).getTime();</script>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7
8 <meta name="robots" content="noarchive" />
9
10 <link rel="alternate" media="handheld" type="text/html" href="http://m.cookpad.com/" />
11 <link rel="apple-touch-icon" href="/images/device/apple-touch-icon.png" />
12
13 <title>レシピ検索No.1／料理レシピ載せるなら クックパッド</title><meta content=",レシピ,簡単,料理,COOKPAD,くっくぱっど,recipe" name="keywords" />
content="日本最大の料理レシピサイト。125万品を超えるレシピ、作り方を検索できる。家庭の主婦が実際につくった簡単実用レシピが多い。利用者は
を公開することもできる。" name="description" />
14
15
16
17
18
19 <script src="/javascripts/jpack.js?1340964911" type="text/javascript"></script>
20
21
22 <script type="text/javascript">AsyncView.load({"specify":["kondate_ext_kondate_unit"],"assigns":"{\\\"kondate_ext_kondate_unit\\\":{}}
,"controller_name":"top","action_name":"top"});</script>
```

ブラウザでHTML表示

右クリックからソースを表示



ブラウザの主な機能

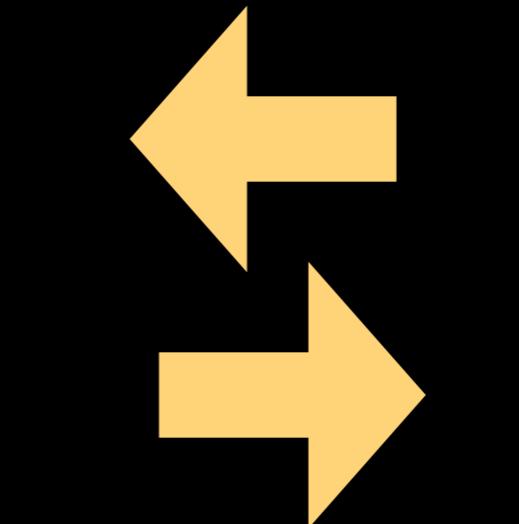
1. リクエストを飛ばす
2. レスポンスでもらったHTMLを人が見る形式で表示する

リクエスト

URL : <http://cookpad.com/>

Web Server

Web App



レスポンス
HTML

Browser

<http://cookpad.com/>



インターネットへのアクセス

Internet

Web Server

Web App

Browser

リクエスト

レスポンス
HTML

WebServerはネットの向こうにある

開発時の構成

自分のPC

Web Server

Web App

URL
リクエスト

Browser

レスポンス
HTML

1台のマシンの中で開発、アクセス可能です。

Rails アプリ

自分のPC

Web Server

Web App
Rails App

URL
リクエスト

レスポンス
HTML

Browser

ブラウザで動作
確認しながら進
めます。



これから作るのはこの部分

では、Railsアプリを
つくりてみましょう

Railsアプリをつくり動かしてみる

1. Railsアプリをつくる
2. WebServerとRailsアプリを起動する
3. ブラウザから以下のURLにアクセスする

1. Railsアプリをつくる

\$ rails new sample_app

```
$ rails new sample_app      [/Users/igarashi/work/0630railssampl
create README.rdoc
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/images/rails.png
create app/assets/javascripts/application.js
create app/assets/stylesheets/application.css
create app/controllers/application_controller.rb
create app/helpers/application_helper.rb
create app/mailers
create app/models
create app/views/layouts/application.html.erb
create app/mailers/.gitkeep
create app/models/.gitkeep
create config
```

...略...

```
Using sass-rails (3.2.5)
Using sqlite3 (1.3.6)
Installing uglifier (1.2.6)
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem
is installed.
```

ちょっと時間がかかります。
途中で**Enter your password to
install the bundled RubyGems to
your system:**
と言わいたらパソコンのパスワードを入力し
てください。

rails new アプリ名
でRailsアプリを作ります。

**Railsが必要なファイルを
たくさんgenerateします。**
**画面に表示されるのはgenerate
されたファイル群です。**

**その後、必要なgemを自動でセッ
トアップします。**

2. WebServerとRailsアプリを起動する

```
$ cd sample_app
```

```
$ bundle exec rails s
```

← s は server の略です。

```
$ bundle exec rails server
=> Booting WEBrick
=> Rails 3.2.3 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2012-06-30 21:32:34] INFO  WEBrick 1.3.1
[2012-06-30 21:32:34] INFO  ruby 1.9.3 (2012-04-20) [x86_64-darwin11.3.0]
[2012-06-30 21:32:34] INFO  WEBrick::HTTPServer#start: pid=2056 port=3000
```

WebServerとRailsアプリを起動して、ブラウザからアクセス可能になります。

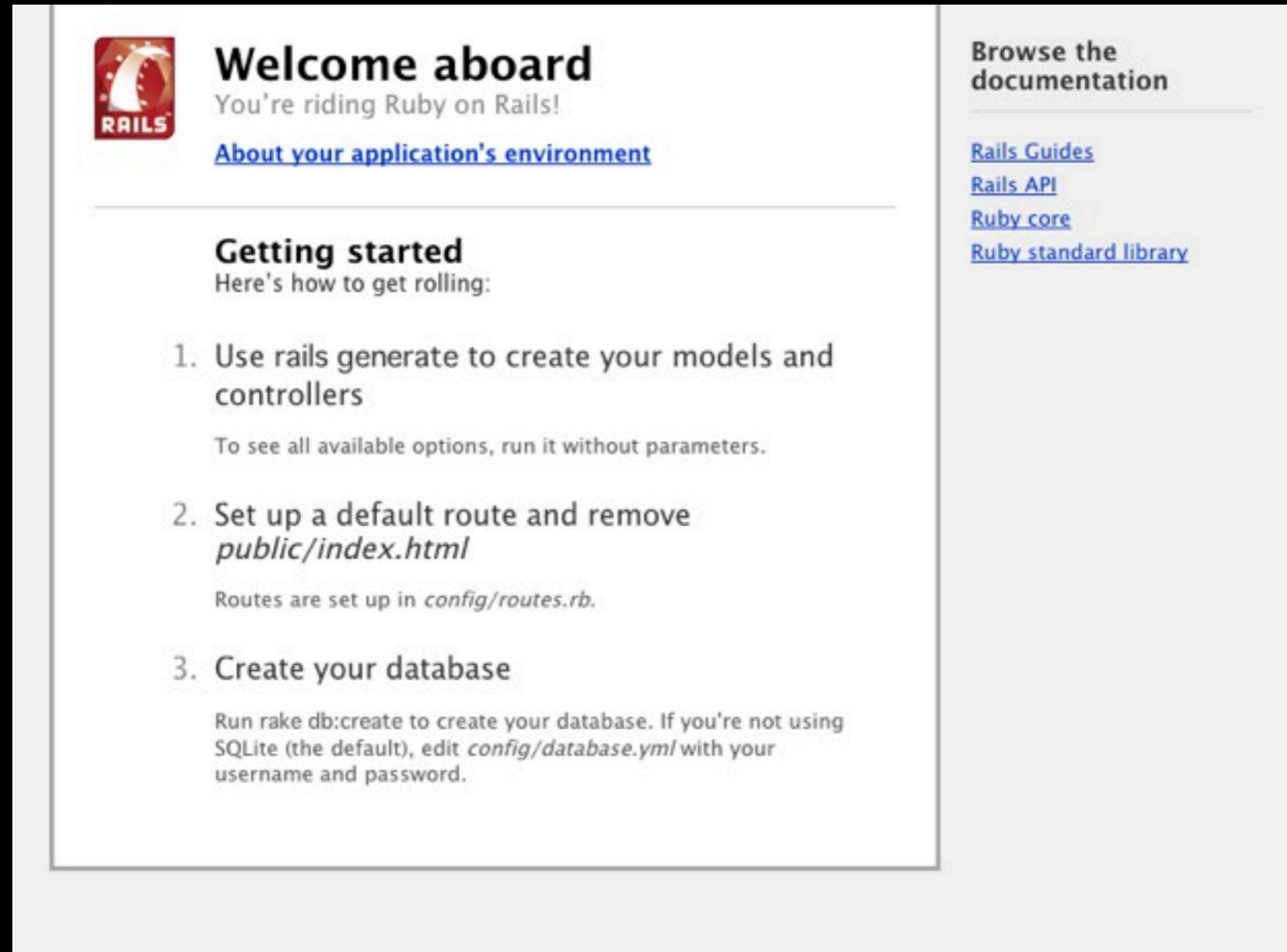
起動中はshellからコマンドなどは入力できないので、必要な場合は別のshell画面を開きます。

終了は Ctrl キー を押しながら c キー を押してください。

(WindowsでCtrl+cで終了しない場合、Ctrl+PAUSEを試してください。)

3. ブラウザから以下のURLにアクセスする

http://localhost:3000



The screenshot shows the initial 'Welcome aboard' screen of a Ruby on Rails application. On the left, there's a red 'RAILS' logo with a white train icon. To its right, the text 'Welcome aboard' is displayed in bold, followed by 'You're riding Ruby on Rails!' in a smaller font. Below this, a blue link reads 'About your application's environment'. A horizontal line separates this from the 'Getting started' section. Under 'Getting started', the text 'Here's how to get rolling:' is followed by three numbered steps: 1. 'Use rails generate to create your models and controllers', with a note below it stating 'To see all available options, run it without parameters.'; 2. 'Set up a default route and remove public/index.html', with a note below it stating 'Routes are set up in config/routes.rb.'; and 3. 'Create your database', with a note below it stating 'Run rake db:create to create your database. If you're not using SQLite (the default), edit config/database.yml with your username and password.' On the right side of the screen, there's a sidebar titled 'Browse the documentation' which includes links to 'Rails Guides', 'Rails API', 'Ruby core', and 'Ruby standard library'.

こんな画面が出ればRailsアプリへアクセス成功です。
開発するための1歩目が踏み出せていることになります。

URLの説明

http://localhost:3000

アドレス ポート

ポートは、同じアドレス内で通信先を特定する番号です。
例えば、アドレスが駅の名前、ポートが駅のホーム〇番線、
とイメージすると分かりやすいかも。

localhost :

特別なアドレスです。自分のマシンを指します。

3000 :

ポート番号です。3000番ポートを指します。

さっきのアクセスのイメージ図

自分のPC
リクエスト
<http://localhost:3000>

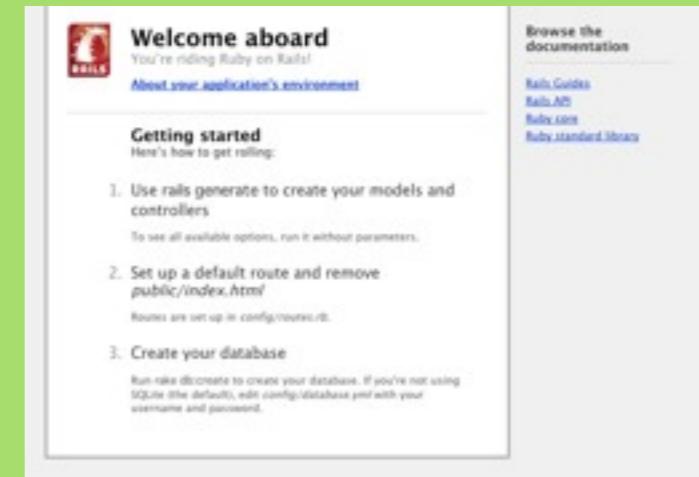
Web Server
localhost:3000

\$ bundle exec rails s

Web App
sample_app

←
レスポンス
HTML

Browser



前につくったRailsアプリを動かしてみる

2回目以降に起動する場合の手順

1. Rails Root フォルダへ移動

sample_appフォルダへ移動しておきます。

sample_appフォルダをRails Root と呼びます。

2. bundle install

```
$ bundle intall
```

Railsアプリで使うgemをセットアップしてくれるおまじない。

ほかのRailsアプリを動かした後などはこの命令が必要です。

3. WebServerとRailsアプリを起動する

```
$ bundle exec rails s
```

4. ブラウザからアクセスする

<http://localhost:3000/books>

つづいて、
簡単なページを作ってみます。

デモ用や開発のためのひな形を
作るためのscaffoldという機
能を利用します。

簡単なページをつくる

本のタイトルとメモを登録できるページをつくります。

sample_appフォルダへ移動しておきます。

(このフォルダのことをRails Root と呼びます。)

1. scaffoldでページをつくる

```
$ bundle exec rails generate scaffold book title:string memo:text
```

2. データをしまうデータベースをつくる

```
$ bundle exec rake db:migrate
```

※ rails server をCtrl-cで停止していたら、また起動する

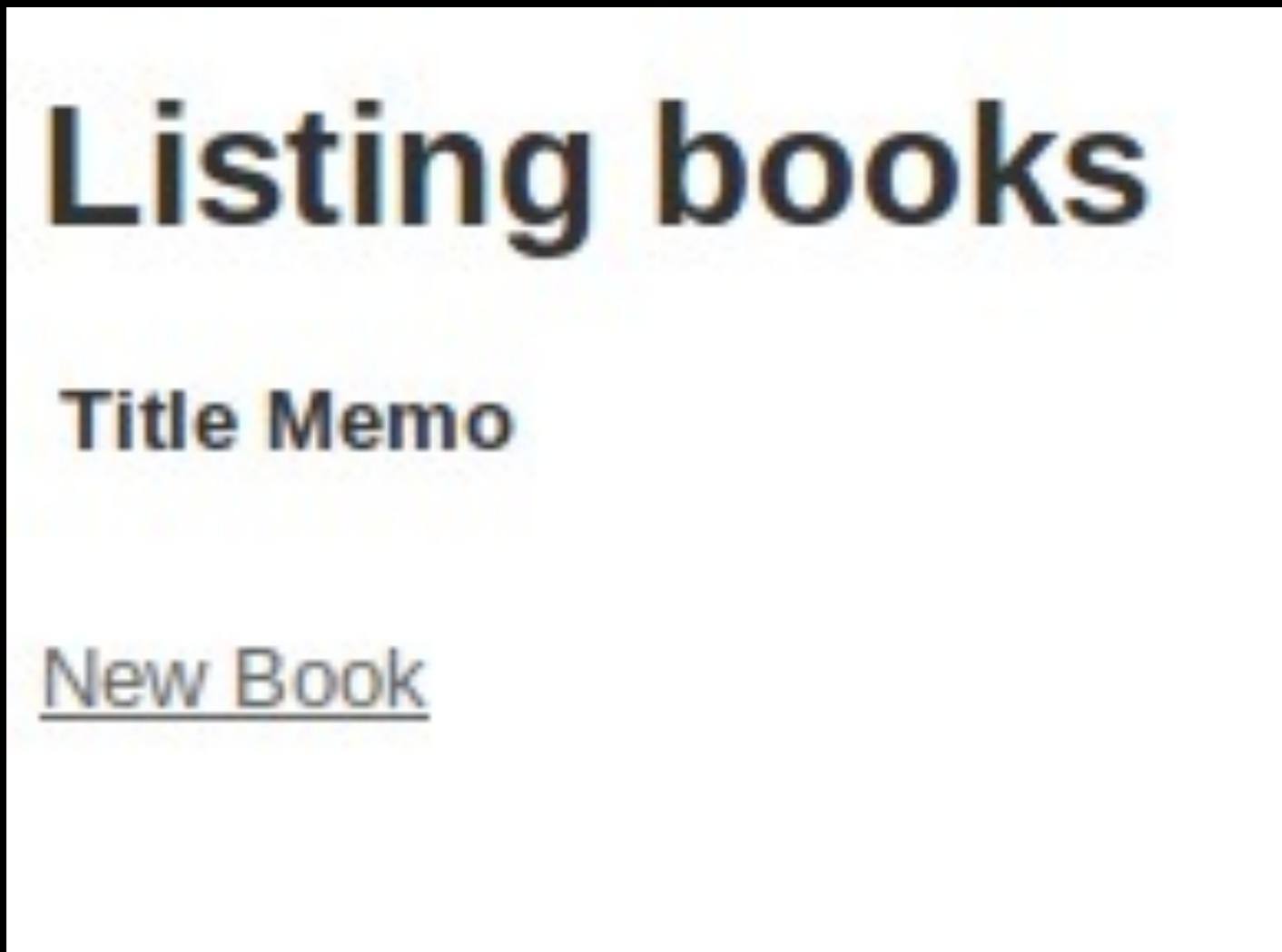
```
$ bundle exec rails s
```

3. ブラウザから以下のURLにアクセスする

```
http://localhost:3000/books
```

ブラウザから以下のURLにアクセスする

http://localhost:3000/books



こんな画面が出れば成功です。 New Book リンクから登録したり編集したり削除したりしてみてください。

では、もう1つ別
のアプリをつくっ
てみましょ。

一番ちいさな
Railsアプリづくり



<http://www.flickr.com/photos/naoyafujii/321375793/>

一番小さなRailsアプリつくり hello world を表示するアプリを作ります。

1. Railsアプリをつくる

```
$ rails new helloworld
```

2. RailsRootへ移動してページのひながたをつくる

```
$ cd helloworld
```

```
$ bundle exec rails g controller hello index
```

g は generate の略です。コマンドの意味は後半で説明します。
打ち間違えて削除したい場合は、g の代わりに d で削除できます。

3. Webサーバを起動する

```
$ bundle exec rails s
```

※終了はCtrl+c (終了しない場合、Ctrl+PAUSEを試してください。)

s は server の略。

4. ブラウザから以下のURLにアクセスする

<http://localhost:3000/hello/index>

4. ブラウザから以下のURLにアクセスする

http://localhost:3000/hello/index

Hello#index

Find me in app/views/hello/index.html.erb

こんな画面が出ればOKです。

この表示を Hello world! に変えてみましょう。

5. Hello world! を表示させる

app/views/hello/index.html.erb

```
<h1>Hello#index</h1>
```

```
<p>Find me in app/views/hello/index.html.erb</p>
```

となっているところを以下のように修正

```
<h1>Hello world!</h1>
```



さきほど同様にブラウザで以下へアクセス

<http://localhost:3000/hello/index>

こんな画面が出ればOKです。

Hello world!

6. ViewでRubyのコードを実行させる

ViewではRubyのコードを実行し、結果を埋め込むことができます。コードを埋め込む場合は`<%=`と`%>`で囲みます。

以下のファイルを変更して、ブラウザからアクセスしてみてください。

`app/views/hello/hello/index.html.erb`

```
<h1>Hello world!</h1>
```

```
<p>現在時刻 <%= Time.now %> </p>
```

`http://localhost:3000/hello/index`

Hello world!

現在時刻 2012-07-10 21:48:56 +0900

← `Time.now` の実行結果

7. ControllerでRubyのコードを実行させる

Viewは表示をする役割を受け持つ部分です。表示する内容を計算したり作成したりするのはControllerなど、Viewの前段で行うしきたりです。以下のファイルを変更して、ブラウザからアクセスしてみてください。

app/controllers/hello_controller.rb

```
class HelloController < ApplicationController
  def index
    @time = Time.now
  end
end
```

app/views/hello/index.html.erb

```
<h1>Hello world!</h1>
<p>現在時刻 <%= @time %> </p>
```

Hello world!

現在時刻 2012-07-10 21:48:56 +0900

<http://localhost:3000/hello/index>

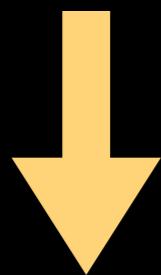
解説

Railsでの 開発の進め方

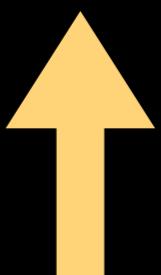
Railsでの開発の進め方

Railsでの基本的な開発の進め方は
以下ののようなサイクルになります。

ひな形になるファイル
(ソースコードや設定ファイル) の生成



この2つの繰り返し



そのアプリ用にソースコードを変更、追記

さきほど作った
アプリも
実はこの流れで
作ってました

一番小さなRailsアプリつくり

hello world を表示するアプリを作ります。

1. Railsアプリをつくる

```
$ rails new helloworld
```

←生成

2. RailsRootへ移動してページのひながたをつくる

```
$ cd helloworld
```

```
$ bundle exec rails g controller hello index
```

←生成

g は generate の略です。コマンドの意味は後半で説明します。

打ち間違えて削除したい場合は、g の代わりに d で削除できます。

3. Webサーバを起動する

```
$ bundle exec rails s
```

※終了はCtrl+c (終了しない場合、Ctrl+PAUSEを試してください。)

s は server の略。

4. ブラウザから以下のURLにアクセスする

```
http://localhost:3000/hello/index
```

5. Hello world! を表示させる コード変更

app/views/hello/index.html.erb

```
<h1>Hello#index</h1>
```

```
<p>Find me in app/views/hello/index.html.erb</p>
```

となっているところを以下のように修正

```
<h1>Hello world!</h1>
```



さきほど同様にブラウザで以下へアクセス

<http://localhost:3000/hello/index>

こんな画面が出ればOKです。

Hello world!

では、どんなものの
が生成されている
のでしょうか。

生成されるフォルダ・ファイル

\$ rails new helloworld

アプリを作るコマンドである

rails new [アプリ名]

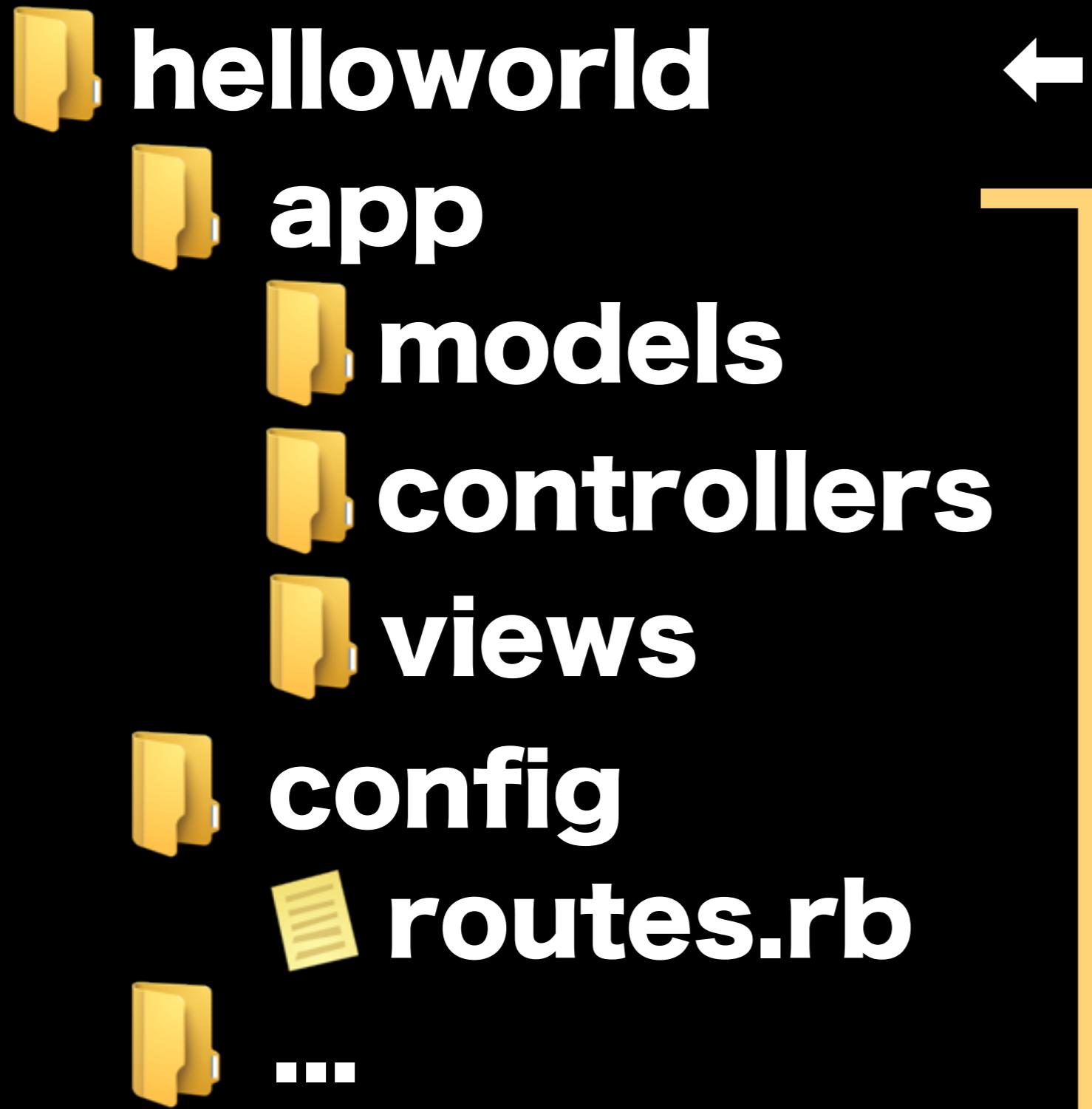
を実行すると、次のページのようなファイル・
フォルダが生成されます。

```
create README.rdoc
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/images/rails.png
create app/assets/javascripts/application.js
create app/assets/stylesheets/application.css
create app/controllers/application_controller.rb
create app/helpers/application_helper.rb
create app/mailers
create app/models
create app/views/layouts/application.html.erb
create app/mailers/.gitkeep
create app/models/.gitkeep
create config
```

←この画面のときに生成
してます。

```
$ rails new helloworld
```

以下のフォルダ・ファイルが生成されます。(一部のみ抜粋)



←アプリ名のフォルダ

Railsアプリ
の基本的な
フォルダと
ファイル群

rails g controller コマンド

```
$ bundle exec rails g controller hello index
```

次は `rails new` に続いて実行した
`rails g controller` コマンドを見てみましょう。

`controller` と `view` などを作る
`rails g controller` コマンドを実行すると
次ページのファイル群が生成されます。
今回のコマンドはURLが `/hello/index` であるペ
ージを表示するためのファイル群を生成します。

```
$ bundle exec rails g controller hello index
```

以下のフォルダ・ファイルが生成・変更されます。(一部のみ抜粋)

 **helloworld**

 **app**

 **models**

 **controllers** ←いろいろな処理を行う箇所

 **hello_controller.rb**

 **views**

←表示を行う箇所

 **hello**

 **index.html.erb**

 **config**

 **routes.rb**

←URLとコントローラの対応表

http://localhost:3000/hello/index

/hello/index ページを
表示するためのファイル群を
生成します。

rails g コマンド

ひながたになるファイルを生成するコマンド

```
$ bundle exec rails g controller hello index
```

generateする種類 アクション名
 コントローラ名

※打ち間違えて削除したい場合は、 g の代わりに d で削除できます。

routes、コントローラ、ビューのファイルほかを生成します。

- ─ config/routes.rb
- ─ app/controllers/hello_controller.rb
- ─ app/views/hello/index.html.erb ほか

※ちなみに、前にやった scaffold もgenerateの種類の1つです。

もう少し詳しく
処理を追いかけ
てみましょう。

ブラウザからアクセスしたときのRailsアプリの動作

リクエスト

- ▶ URL : `http://localhost:3000/hello/index`
- ▶ HTTPメソッド : GET

Web Server

Rails App

Routes

Controller

View



リクエスト



レスポンス HTML

Browser

URL入力欄に
アドレスを入力して
Enterを押すと
リクエストが飛びます

Hello#index

Find me in app/views/hello/index.html.erb

Railsアプリがリクエストを受けて レスポンスを返すまでの流れ

Routes, Controller, View の3つの部品を通過します。

リクエスト

- ▶ URL : `http://localhost:3000/hello/index`
- ▶ HTTPメソッド : GET

Rails App

Routes

routes.rb

Controller

hello_controller.rb indexアクション

View

hello/index.html.erb

レスポンス : HTML

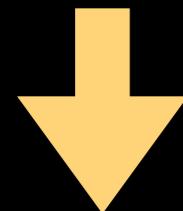
Routes

リクエストのURLとHTTPメソッドに応じて
処理を行う先を決めます。

詳しくはまた後で出てきたときに説明していきます。

リクエスト

- ▶ URL : `http://localhost:3000/hello/index`
- ▶ HTTPメソッド : GET



routesが持っている対応表を見て呼び出すアクションを決定

HelloControllerのindexアクション

※routesからアクセスされるコントローラのメソッドのことを
「アクション」と呼びます。

Controller

さまざまな処理を行い、つぎの View に処理を渡します。

HelloController の ソースは
app/controllers/hello_controller.rb です。

app/controllers/hello_controller.rb

```
class HelloController < ApplicationController
  def index
    @time = Time.now
  end
end
```

ポイントは、Viewにデータを渡す際は、インスタンス変数(@はじまりの変数)を使うことです。ここでは、@time が次のViewへ渡されます。

→次はView(app/views/hello/index.html.erb)に処理が渡ります。

View

ユーザーの目に届く部分（ここではHTML）をつくります。

app/views/hello/index.html.erb

```
<h1>Hello world!</h1>
<p>現在時刻 <%= @time %> </p>
```

index.html.erb は、HTMLのもとになるファイルです。Rubyのコードを実行した結果を埋め込むことができます。（このようなファイルをテンプレートと呼びます。）埋め込まれたRubyのコードを実行した結果でHTMLを作ります。

Hello world!

現在時刻 2012-07-10 21:48:56 +0900

← Controllerで代入した
@time の中身が表示される

（実際には、Viewテンプレートファイルから作られたHTMLに、Railsがその他の加工を加えてレスポンスとして送出します。）

Controllerでコードを実行して、結果をViewへ渡すときはインスタンス変数を使う

リクエスト

- ▶ URL : `http://localhost:3000/hello/index`
- ▶ HTTPメソッド : GET

Rails App

Routes

Controller コードを実行して`@time`に代入

結果を渡すにはインスタンス変数(`@はじまり`)を使う

View

`@time`の中身を表示する

レスポンス : HTML

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