

Ruby 講義

第13回

Sinatraで作る簡単Webアプリ

五十嵐邦明

twitter : igaiga555

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



2013.7.11 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

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>

先週の復習

★TODO

ここから今週の内容

Sinatra

Put this in
your pipe

[README](#)
[DOCUMENTATION](#)
[CONTRIBUTE](#)
[CODE](#)
[CREW](#)
[ABOUT](#)

require 'sinatra'

```
get '/hi' do  
  "Hello World!"  
end
```

and smoke it
Sinatra で作る簡単Webアプリ

```
$ gem install sinatra  
$ ruby hi.rb
```

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 | 口コミ

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 ふりふりふりん

・【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



家族が笑顔になる！

我が家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任せ
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



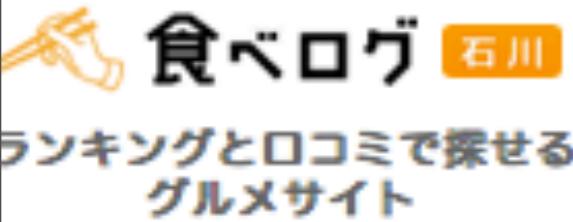
あっという間に一品完成



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



TOEIC®テスト 1日1分
達人の極意♪ 明日



教えてあげる!
ハイスコア獲得の秘密

エリア・駅 石川県

キーワード 津幡

検索

検索ヘルプ

初めての方へ (食べログとは?)

グルメ・レストラン > 石川グルメ > 検索結果

6,787 店舗 | 18,749 口コミ

グルメ検索

エリアを絞り込む

石川県

金沢

加賀・白山

能登南部

能登北部

石川県の路線・駅から探す

石川県の市区町村から探す

ジャンルを絞り込む

すべて

レストラン

カフェ・喫茶

パン・スイーツ

バー・お酒

旅館・オーベルジュ

その他

ジャンル一覧から探す

石川県の津幡に関連するお店を探す



ツイート

いいね!

「石川県」「津幡」で検索しました。 [検索条件を変更する](#)

お得な《津幡駅》宿泊なら

www.jalan.net

じゃらんなら憧れのあの宿も、格安のあの宿も、宿泊プラン多数ご用意!

-PR-

★ 点数順

人気順

最近の注目順

こだわりで並び替え

お店の点数が高い順に表示しています。

[点数とは?](#)

並び替え

点数が高い順

夜の点数が高い順

昼の点数が高い順

すべて

店名で絞り込み

口コミで絞り込み

店名に「津幡」を含むレストラン **1~20件** を表示 / 全21件

1 回転寿司海王 津幡店 (津幡 / 回転寿司)

[ブックマークに追加](#)



★★★★★ 3.05

3.00

3.06

[口コミ: 3件]

~¥999

~¥999

飲食

料理・味 3.05 | サービス 3.00 | 雰囲気 2.97 | CP 3.05 |

酒・ドリンク 3.00

料金: 5,098円 (税込) 5,184元 (税込) (10/09/05 初投稿)

食べログ

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
52,500円



シモンズ 14-1055-
10-1055-
52,750円



ディオール D041-103
52,750円



59 ヒステリック
CHAPLIN-gold
52,500円

Oh my Glasses

Rails製



Twitterへようこそ

周りの人々や、興味のある企業や団体などの組織
で、今何が起きているのかを見つけましょう。

ユーザー名、またはメールアドレス

パスワード

ログイン

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に登録する

では、早速ですが、
Webアプリをつくっ
てみましょう

sinatra gem を使ったWebアプリ作り演習

1.spreadsheet gem をインストールします。

\$ gem install sinatra

2.以下のコードを書いて好きな名前で保存します。 (例えばserver.rb)

```
require "sinatra"
```

```
get "/" do  
  "Hello world!"  
end
```

3.実行します。右下のような表示が出ます。

\$ ruby server.rb

```
== Sinatra/1.4.2 has taken the stage on 4567 for  
development with backup from Thin  
>> Thin web server (v1.5.1 codename Straight Razor)  
>> Maximum connections set to 1024  
>> Listening on localhost:4567, CTRL+C to stop
```

4.ブラウザから以下のURLへアクセスします。

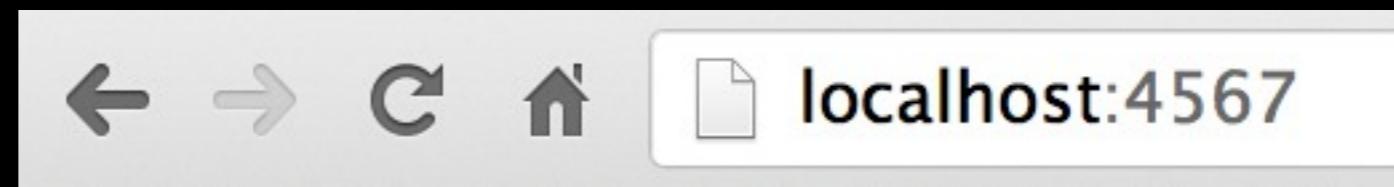
<http://localhost:4567>

"Hello world!"と表示されれば成功です。

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

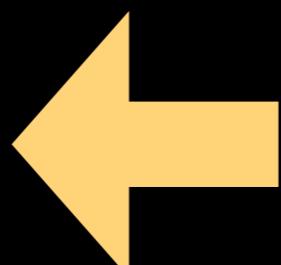
URL : <http://localhost:4567/>

Web Server

Web App

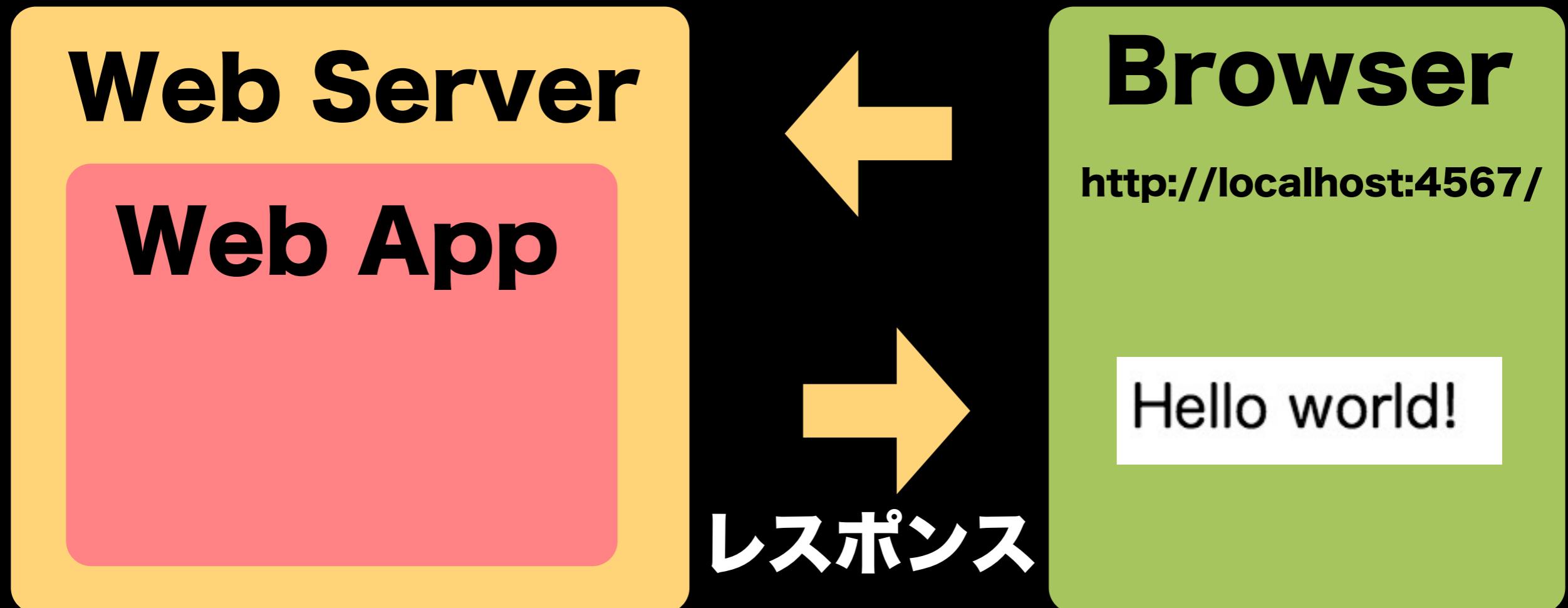
Browser

<http://localhost:4567/>



リクエストが飛びます。

Webアプリの基本動作



レスポンスが返ってくる

レスポンスを
ブラウザが表示

インターネットへのアクセス

Internet

Web Server

Web App

リクエスト

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



レスポンス

Browser

<http://cookpad.com/>



WebServerは通常はネットの向こうにある

どのWeb Server へアクセスするかは リクエストのURLで決まる

Web Server

Web App

リクエストのURLが <http://cookpad.com> の場合は、
クックパッドさんの運営するWeb Server へ繋がります。
URLはインターネット上の住所みたいなものです。

Internet

リクエスト

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



レスポンス

Browser

 http://cookpad.com



先ほど作ったアプリの場合は
どうなっているかと言うと

URL localhost は自分のマシン localhost は自分のPCを表す特別なアドレスです。

自分のPC

Web Server

Web App

```
require "sinatra"  
  
get "/" do  
  "Hello world!"  
end
```

リクエスト

URL : <http://localhost:4567/>

レスポンス

Browser

localhost:4567

Hello world!

Web App(アプリ)の部分がみんなさんが書いたコードです。
(Web Server はsinatra とRubyがうまいことやってくれます。)

では、URLの:より後の 4567 とは何でしょうか？

URLの説明

http://localhost:4567

アドレス ポート

URLの : より後ろはポートを表します。

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○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。

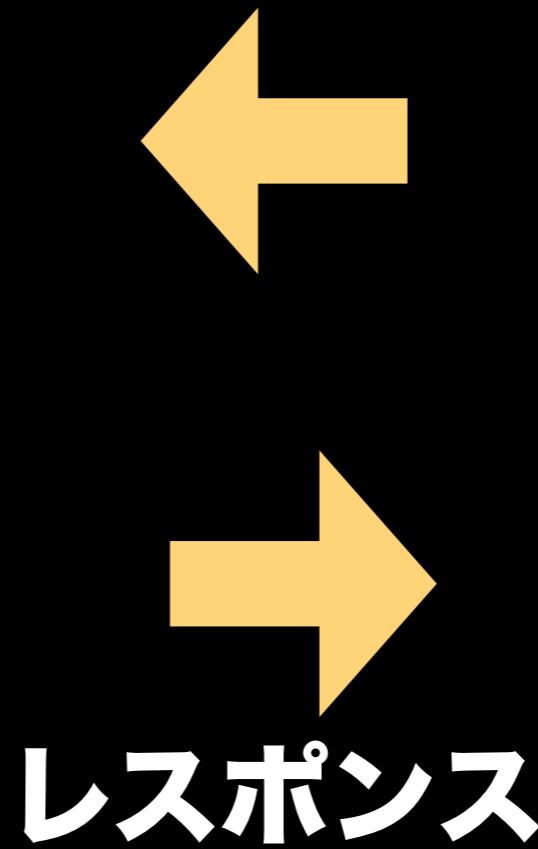
4567 :

ポート番号です。4567番ポートを指します。

先ほど作ったWebアプリの動作概略



リクエスト
URL : <http://localhost:4567/>



次はWebアプリの動きを詳しく見ていきます

Web Server

localhost 4567番ポートに来た
リクエストをWebAppへ届けます

Web App

URLのアドレスとポートより
後ろの部分をパスと言います。
この場合、パスは / になります。

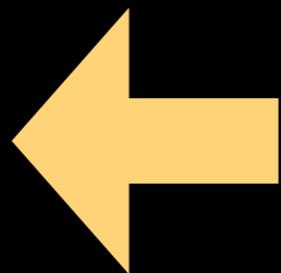
```
require "sinatra"  
  
get "/" do  
  "Hello world!"  
end
```

パスが / のリクエストに対して
"Hello world!"というレスポンス
を返すWebアプリとして動作します

Webアプリの動作

リクエスト

URL : <http://localhost:4567/>



Browser

← → C ⌂ localhost:4567

Hello world!

受け取ったレスポンスを表示

レスポンス

sinatra gem を使ったWebアプリ作り演習

前の手順で実行したWeb サーバーが起動している場合は以下の手順で停止します

★TODO

2.以下のコードを書いて好きな名前で保存します。 (例えばserver.rb)

```
require "sinatra"
```

```
get "/" do  
  "Hello world!"  
end
```

3.実行します。右下のような表示が出ます。

\$ ruby server.rb

```
== Sinatra/1.4.2 has taken the stage on 4567 for  
development with backup from Thin  
>> Thin web server (v1.5.1 codename Straight Razor)  
>> Maximum connections set to 1024  
>> Listening on localhost:4567, CTRL+C to stop
```

4.ブラウザから以下のURLへアクセスします。

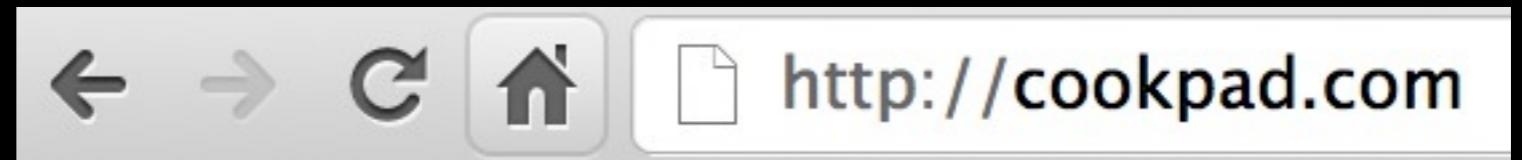
<http://localhost:4567>

"Hello world!"と表示されれば成功です。

TODO

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

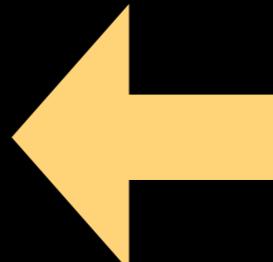
URL : http://cookpad.com/

Web Server

Web App

Browser

http://cookpad.com/



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

Browser

レスポンス
HTML

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

Webアプリ

Rubyのライブラリ管理システム

目次

Gem

Fragrant from
my work
With love to you

Hector



Gem

Rubyのライブラリ管理システム

Gem

ライブラリ：
他のプログラムから読み込んで利用するためのプログラム

たくさんの便利なライブラリがネット上に公開されています。それらをインストールするなど、管理してくれる仕組みがGemです。

Gemの例

- Excelファイルを読み書きする
 - twitterなど有名なWebサービスへのアクセスを簡単にしてくれる
 - 画像をリサイズしたりエフェクトをかける
- などなど数万種類のライブラリが公開されています

gemを紹介、管理しているサイト

**710,437,847
downloads**

of 40,705 gems cut since July 2009

Welcome to your community RubyGem h

Find your gems easier, publish them faster, and hav

do

LEARN

Install RubyGems 1.8.24

Ruby's premier packaging system

Browse the Guides

In depth explanations, tutorials, and references

Gem Specification

Your gem's interface to the world

SHARE

`gem update --system`

Update to the latest RubyGems version

`gem build foo.gemspec`

Build your gem

`gem push foo-1.0.0.gem`

Deploy your gem instantly

RubyGems.org is the Ruby community's gem hosting service. Instantly publish your gems and install them. Use the API to interact and find out more information about available gems. Become a contributor and enhance the site with your own changes.

<http://rubygems.org>

Gemの使い方

gemパッケージのインストール

1. shell からgemコマンドを使ってインストール

```
$ gem install gem名
```

2. コードで require "gem名"

```
require "gem名"  
# ここでgem を使うコードを書きます。
```

Gemの使い方

インストール済のgemパッケージ一覧を表示

shell からgem list コマンドで一覧表示

```
$ gem list
```

spreadsheet

Excelファイルを操作するgem

spreadsheet

Excelファイルを読み書きするgem
Excelがインストールされていなくても、
Windows以外でも動作します。

shell から以下のコマンドを実行するとインストー
ルできます。

```
$ gem install spreadsheet
```

spreadsheet gem 演習

spreadsheet gem をインストールしてから実行してください。

\$ gem install spreadsheet

a1. Excelファイルを新規作成

```
require "spreadsheet"
```

```
book = Spreadsheet::Workbook.new
sheet = book.create_worksheet
sheet[0,0] = "testing..."
book.write("example.xls")
```

※**Spreadsheet::Workbook**
は、 **Spreadsheet**モジュールの中の
Workbookクラスの意味です。先週
説明したモジュールにはこのように種
類を分けて名前をつける使い方あり
ます。また次回以降の講義で出てきた
際に説明します。

生成した**example.xls**を、 Excelか、後述のオープンソースオフィスアプ
リで開いてみてください。

動作したら、 **sheet[0,0]** の **0,0** の部分を変えてどこに文字列が記入され
るか見てみてください。

spreadsheet gem 演習

a2. 既存のExcelファイルを開いて編集して別名で保存

```
require "spreadsheet"
```

```
book = Spreadsheet.open("example.xls")
sheet = book.worksheet(0)
p str = sheet[0,0]
sheet[1,0] = "hello"
book.write("example2.xls")
```

example.xlsを開き、画面に内容を表示したり、
セルに値を追記して別名**example2.xls**として保存するサンプルです。
Excelファイルのデータを解析するような場合や、
Excel形式のデータを作る場合に利用できます。

Office Application

Libre Office

<http://www.libreoffice.org/download>

Apache Open Office

<http://www.openoffice.org/ja/download/>

無料で使えるExcelファイルが読み書きできるアプリ
Win/Mac/Linux 用がそれあります。

Twitter

twitterを検索したり操作したりするgem

Twitter

A Ruby interface to the Twitter API.

This project is maintained by [sferik](#)

 Download ZIP

 Download TAR

 View On GitHub

The Twitter Ruby Gem

gem version 4.8.1

build passing

dependencies up-to-date



A Ruby interface to the Twitter API.

Installation

```
gem install twitter
```

To ensure the code you're installing hasn't been tampered with, it's recommended that you verify the signature. To do this, you need to add my public key as a trusted certificate (you only need to do this once):
<http://sferik.github.io/twitter/>

twitter gem 演習

twitter gem をインストール

\$ gem install twitter

twitterでユーザーがつぶやいている「最近聞いている曲」を調べる
nowplaying という言葉入りのtweetを最新10件検索する

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

コードを順番に説明していきます。

twitter 認証キー

最初の Twitter.configure ブロックはtwitter開発者サイトで取得する認証キーを設定します。

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

twitter API(情報取得操作窓口)を利用するためには、
twitterへ登録して許可をもらう必要があります。
そのための鍵が認証キーです。

twitter 認証キー 取得

以下のtwitter開発者用ページへログインします。

June 11, 2013 Having trouble with your app? API v1 is retired and no longer functional. | Read more →

 Developers

API Health Blog Discussions Documentation

Search



Sign in

Home

Sign in with your Twitter account

Please log in to access that page.

Username: *

igaiga555

New to Twitter? [Sign up](#)

Password: *

Log in

<https://dev.twitter.com/apps/new>

twitter 認証キー 取得

Application Details

Name: *
lec2013

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: *
Lecture 2013

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website: *
<http://example.com>

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For [@Anywhere applications](#), only the domain specified in the callback will be used. [OAuth 1.0a](#) applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

アプリ(これから作る
プログラム)の情報を
入力します。

* 印の必須項目を埋め
ます。 Website欄は
このアプリの情報を記
載するページのURL
を書く欄ですが、ペー
ジがまだない場合は説
明文に従って仮で埋め
ておきます。

twitter 認証キー 取得

Developer Rules Of The Road

Last Update: September 5, 2012.

Rules of the Road

Twitter maintains an open platform that supports the millions of people around the world who are sharing and discovering what's happening now. We want to empower our ecosystem partners to build valuable businesses around the information flowing through Twitter. At the same time, we aim to strike a balance between encouraging interesting development and protecting both Twitter's and users' rights.

So, we've come up with a set of Developer Rules of the Road ("Rules") that describes the policies and philosophy around what type of innovation is permitted with the content and information shared on Twitter.

The Rules will evolve along with our ecosystem as developers continue to innovate and find new, creative ways to use the Twitter API, so please check back periodically to see the current version. Don't do anything prohibited by the Rules and talk to us if you think we should make a change or give you an exception.

If your application will eventually need more than 1 million user tokens, or you expect your [embedded Tweets](#) and [embedded timelines](#) to exceed 10 million daily impressions, you will need to talk to us directly about your access to the Twitter API as you may be subject to additional terms.

Furthermore, applications that attempt to replicate Twitter's core user experience (as described in Section I.5 below) will need our permission to have

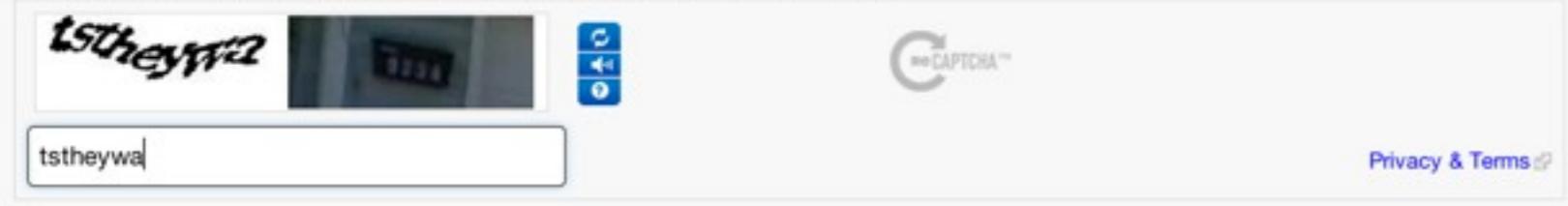
Yes, I agree

By clicking the "I Agree" button, you acknowledge that you have read and understand this agreement and agree to be bound by its terms and conditions.

利用規約に同意して
チェックします。

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.



いたずら防止機能である
CAPTCHAに文字を入
力します。(画像に書かれ
ている読みづらい文字を
入力)

全て入力したら "Create your Twitter application" ボタンを押します。

twitter 認証token 取得

lec2013

Details Settings OAuth tool @Anywhere domains Reset keys Delete

 Lecture 2013
<http://example.com>

Organization
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

OAuth settings
Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level	Read-only About the application permission model
Consumer key	e1uqYcOjHqgtFB1NRD2l0Q
Consumer secret	1o2t0fd29Ns8WJtvwacHHH0xlznC4mhfQF3erKL44g
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token
Callback URL	None
Sign in with Twitter	No

Your access token
It looks like you haven't authorized this application for your own Twitter account yet. For your convenience, we give you the opportunity to create your OAuth access token here, so you can start signing your requests right away. The access token generated will reflect your application's current permission level.

[Create my access token](#)

登録すると
Consumer key
Consumer secret
が得られます。
メモしておきましょう。

※注意：
Consumer secretは
秘密にする情報なので
←こんな風に公開しちゃいけません。

次は
Create my access token
ボタンを押します。

twitter 認証token 取得

しばらくするとAccess token, Access token secret
が表示されます。メモしておきます。

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign requests with your own Twitter account. Do not share your oauth_token_secret with anyone.

Access token 4596601-E2E7bNKVy78kDSrGWnlzFhPjPfa55XbXaPR97FPsOY

Access token secret 8aSTcKaksAC4kaP4x9Tz8KBKuCLInH5kEqARoK41YY

Access level Read-only

[Recreate my access token](#)

※注意：Access token secret は秘密にする情報なので
こんな風に公開しちゃいけません。

トークンについて解説

トークン：
硬貨の代わりに用いられる代用貨幣のこと。

- Wikipedia

4つのトークン

公開

公開トークンは
id のように使う

非公開

非公開トークンは
password の
ように使う

Consumer key

Consumer secret

Consumer key/secret

Access token

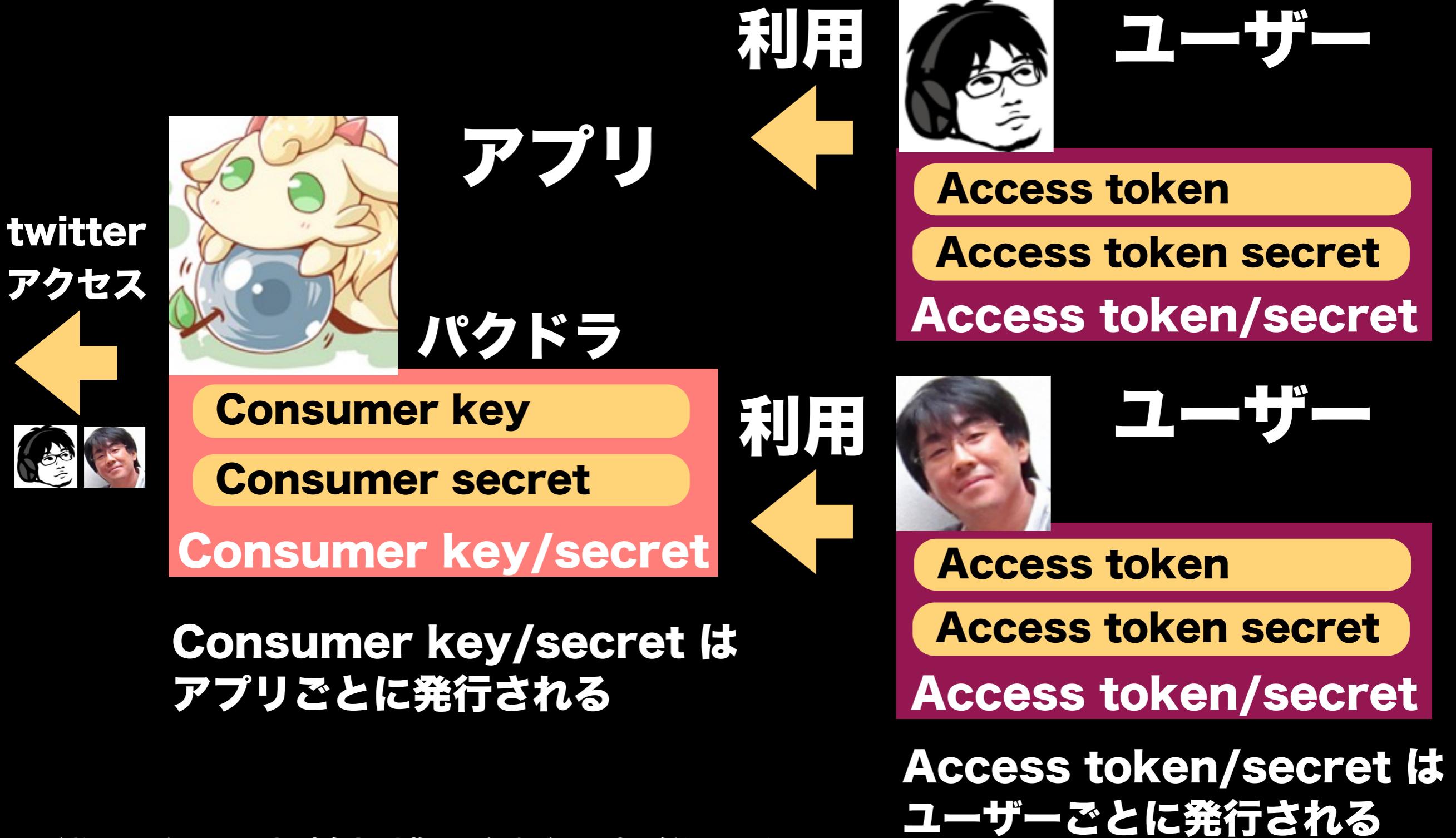
Access token secret

Access token/secret

ここで言うConsumerはアプリのことです。
例えば、あるアプリに悪意があることが分かり、
そのアプリについては認証させないようにする場合
はこの2つのトークンを無効にすれば遮断できます。

ここで言うAccess tokenは利用
者1人1人に発行されます。利用者
がこのアプリに対する認証を禁止す
る場合には、この2つのトークンを
無効にします。

トークンとアプリ、ユーザーの関係



※この例えはフィクションです。実在する人物、アプリなどには一切関係ありません。

アプリを利用不可にする場合

利用不可にするアプリの Consumer key/secret を無効化します。
結果的に、そのアプリを使っているユーザー全員が利用不可になります。

twitter
アクセス



アプリ

パクドラ

Consumer key



Consumer secret



Consumer key/secret

Consumer key/secret は
アプリごとに発行される

利用



ユーザー



Access token

Access token secret

Access token/secret

利用



ユーザー



Access token

Access token secret

Access token/secret

Access token/secret は
ユーザーごとに発行される

※この例えはフィクションです。実在する人物、アプリなどには一切関係ありません。

あるユーザーのみアプリを利用不可にする場合

自分の Access token/secret を無効化します。

自分だけはそのアプリがtwitterへアクセスするのを不可能にできます。

twitter
アクセス



アプリ

パクドラ

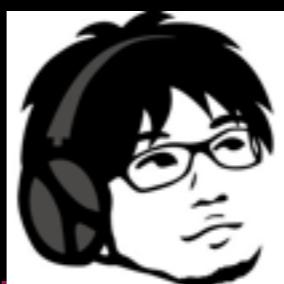
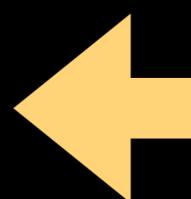
Consumer key

Consumer secret

Consumer key/secret

Consumer key/secret は
アプリごとに発行される

利用



ユーザー

Access token

Access token secret

Access token/secret

利用



ユーザー

Access token

Access token secret

Access token/secret

Access token/secret は
ユーザーごとに発行される

※この例えはフィクションです。実在する人物、アプリなどには一切関係ありません。

では、コードに
戻りましょう

twitter 認証キー

twitterサイトから取得した4つの値を対応する変数に設定します。

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp =>10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

twitter 認証キー

次は後半部分を説明します。

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

twitter情報取得

twitterでユーザーがつぶやいている「最近聞いている曲」を調べる
nowplaying という言葉入りのtweetを最新10件検索する

```
この言葉を含む 件数  
Twitter.search("nowplaying", :rpp => 10,  
  :result_type => "recent").each do |status|  
  puts "#{status.from_user}: #{status.text}"  
end      つぶやいたユーザー  つぶやき内容
```

Twitter.searchメソッドを使って検索します。
引数に検索したい内容を渡します。
10件取得したつぶやきは、1件ずつブロックが実行され、
ブロック内変数 status に代入されます。

演習解説 文字列表示

```
puts "#{status.from_user}: #{status.text}"
```

"#{変数名}" と書くと、文字列に変数の内容を埋め込むことができます。

```
name = "igarashi"
```

```
puts "Author #{name}" #=> Author igarashi
```

変数nameの中身が表示される

以下の2つのコードは同じ動作です。

```
name = "igarashi"
```

```
text = "Author #{name}"
```



```
text = "Author igarashi"
```

同じ動作

演習解説 文字列表示

Rubyには2種類の文字列の表現方法があります。

ダブルクオート（"）とシングルクオート（'）です。

前のページで説明した変数の中身を表示する場合はダブルクオートを使う必要があります。

ダブルクオート："#{変数名}" と書くと変数の中身を表示

```
name = "igarashi"  
puts "Author #{name}" #=> Author igarashi
```

変数nameの中身が表示される

シングルクオート：'#{変数名}' と書くとそのまま表示

```
name = "igarashi"  
puts 'Author #{name}' #=> Author #{name}
```

変数nameの中身が表示されず、そのまま出力される

演習解説 引数のHashの省略形

この部分、見慣れない書き方をしています。

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")
```

省略しないで書くと以下のようになります。

```
Twitter.search("nowplaying", {:rpp => 10, :result_type => "recent"})
```

1番目の引数は文字列 2番目の引数はHashで[]を省略

Rubyでよく使われるテクニックで、たとえば複数のオプションを引数として渡したいときにHashを渡すことがあります。キーが説明文の役目をしてくれて読みやすいからです。

その際に曖昧にならなければHashの[]を省略して書けます。

twitter gem 演習

では実行してみましょう！

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

※Windowsで以下のようなエラーが出る場合は次ページ参照

```
C:/RailsInstaller/Ruby1.9.3/lib/ruby/1.9.1/net/http.rb:799:in
`connect': SSL_connect returned=1 errno=0 state=SSLv3 read server
certificate B: certificate verify failed (Twitter::Error::ClientError)
```

Windows で以下のようなエラーが出る場合の対応方法

```
C:/RailsInstaller/Ruby1.9.3/lib/ruby/1.9.1/net/http.rb:799:in `connect':  
SSL_connect returned=1 errno=0 state=SSLv3 read server certificate  
B: certificate verify failed (Twitter::Error::ClientError)
```

Rubyが使うSSL証明書(セキュリティのために必要)の期限が切れたために発生する問題です。

1. 以下から証明書ファイルをDLして

C:¥RailsInstaller¥cacert.pem という名前で保存します。

<http://curl.haxx.se/ca/cacert.pem>

2. 以下のコマンドを打ち、環境変数 **SSL_CERT_FILE** にそのパスを設定

\$ setx SSL_CERT_FILE "C:¥RailsInstaller¥cacert.pem"

3. 開いているコマンドプロンプトは一度閉じて開きなおす

これでエラーが出なくなります。

twitter gem 演習

```
require "twitter"
```

```
Twitter.configure do |config|
  config.consumer_key = "XXXXXXXXXXXXXXXXXXXX"
  config.consumer_secret = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token = "XXXXXXXXXXXXXXXXXXXX"
  config.oauth_token_secret = "XXXXXXXXXXXXXXXXXXXX"
end
```

```
Twitter.search("nowplaying", :rpp => 10, :result_type => "recent")[:statuses].each do |status|
  puts "#{status.from_user}:#{status.text}"
end
```

検索語"nowplaying"の部分をいろいろ変えて実行してみてください。
(twitter API は単位時間内に実行できる回数が限られているので、実行しすぎると1時間程度実行できなくなります。)

日本語にする場合はマジックコメント(# coding:utf-8)を忘れずに。
また、Windowsで結果が文字化けする場合は、以下のようにファイル
result.txt へ書き出してエディタで開く、を試してみてください。

```
$ ruby twitter.rb > result.txt
```


參考資料

書式

Rubyコード

`puts "abc"`

実行結果

"abc"

実行結果

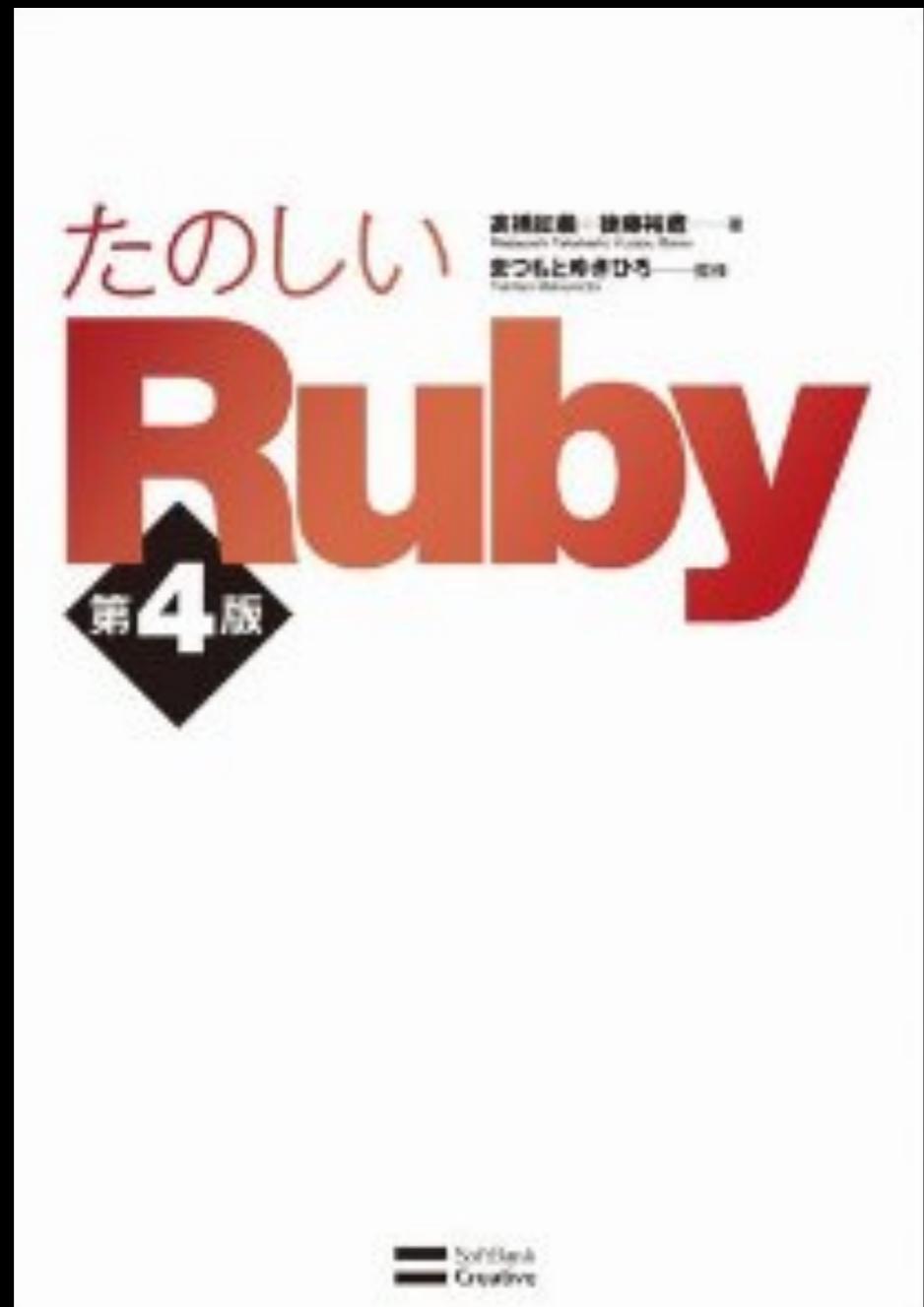
文中では `#=>` で書きます

`p 1+2 #=> 3`

shellコマンド

`$ ls`

教科書：たのしいRuby



<http://www.amazon.co.jp/dp/4797372273/>



お買い求めは
大学生協または
ジュンク堂池袋店で

課題チェック用の付箋の書き方

上の方に学籍番号と名前を書いてください

学籍番号

名前

※この辺に講師陣がクリアした課題番号を書いていきます。

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