



spice life

**Kuniaki IGARASHI**

**2016.7.23 Rails Girls 東京**



**五十嵐邦明**  
**株式会社 spice life**  
**CTO**

**github: igaiga**  
**twitter: igaiga555**  
**facebook: kuniaki.igarashi**

# TMIX

## Rails製



0120-570-150  
月～土 9:00～18:00(祝日除く)



カート



会員登録



ログイン

アイテムと価格

デザイン

法人・物販

TMIXについて

ショールーム

新着情報



## 作れば絶対楽しくなる

オリジナルTシャツが1枚から簡単に作れます

 作ってみる





# spice life sponsored events



# RailsGirls スポンサー



Kobe  
2015/09/11-12



Tokyo  
2015/09/11-12



Toyama  
2015/08/25



Fukuoka  
2015/07/24-25



Osaka  
2015/06/19-20



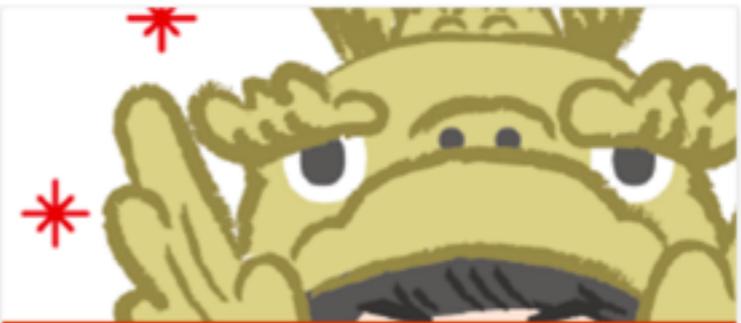
Kyoto  
2015/04/24-25



Shiojiri  
2015/03/07



Kyoto  
2014/11/28-29



Nagoya  
2014/10/17-18

# 地域Ruby会議スポンサー

## 地域Ruby会議

未確定情報を含む、今後の開催情報は、日本Ruby会の公式Wikiの[これから開催される地域Ruby会議](#)

Date	Name	Report
2015-11-08(Sun)	<a href="#">大江戸Ruby会議05</a>	レポートはまだありません
2015-09-26(Sat)	<a href="#">松江Ruby会議07</a>	レポートはまだありません
2015-08-29(Sat)	<a href="#">TokyuRuby会議09</a>	レポートはまだありません
2015-07-11(Sat)	<a href="#">関西Ruby会議06</a>	レポート
2015-06-20(Sat)	<a href="#">とちぎRuby会議06</a>	レポート
2015-03-28(Sat)	<a href="#">浜松Ruby会議01</a>	レポート
2015-01-17(Sat)	<a href="#">神奈川Ruby会議01</a>	レポート



**RubyKaigi 2015**

# TMIX

- 1枚からつくれます
- 色数無制限
- 即日発送商品あり
- 送料無料
- Tシャツ2490円～
- パーカー, トートバックほか
- 割引きクーポンあります

Print Your Heart



+



=



COTTON

HEART

TMIX

# TMIXラインナップ



半袖Tシャツ | レディースTシャツ | ポロシャツ | 長袖Tシャツ | スウェット | パーカー | ベビー・子供服 | エコバッグ | エプロン | タオル

オリジナルTシャツ HOME > RailsGirlsKyoto3のデザイン

### RailsGirlsKyoto3

デザインID : 2092736  
myページからコメントを編集できます

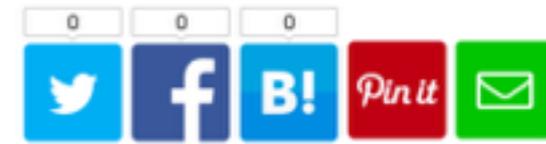


このデザインを購入する

オリジナルTシャツ HOME > esa 0Θ0 和 ショク ミート のデザイン

### esa 0Θ0 和 ショク ミート

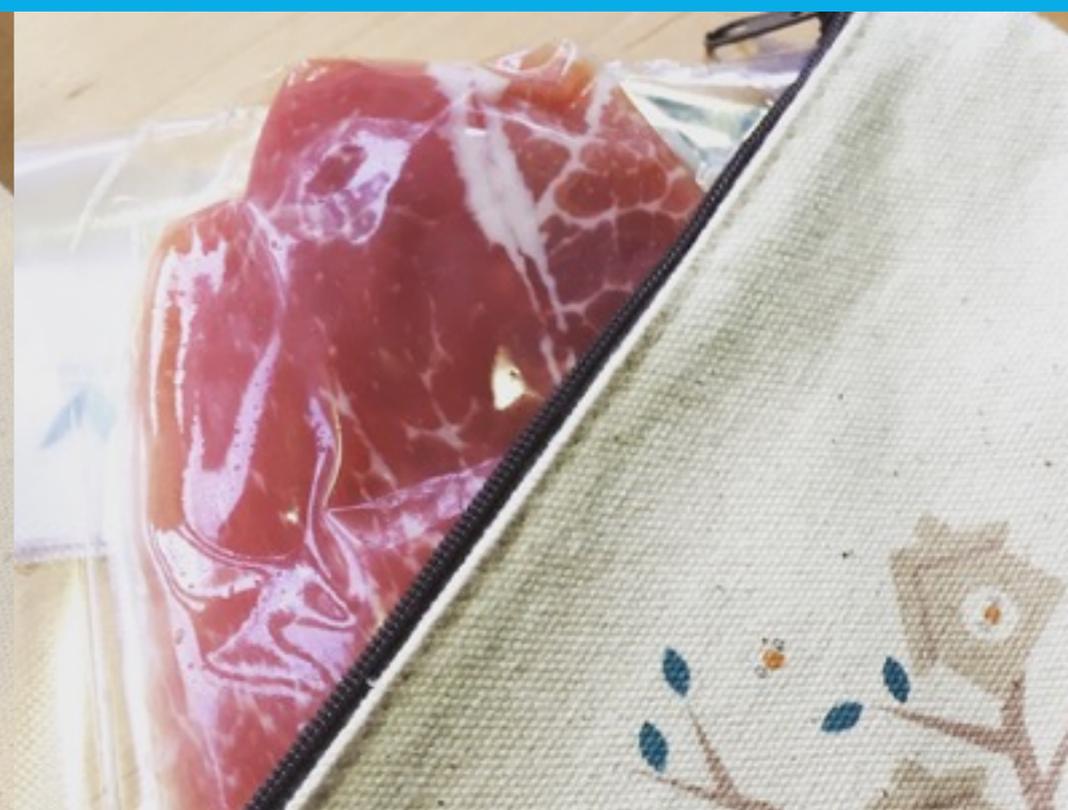
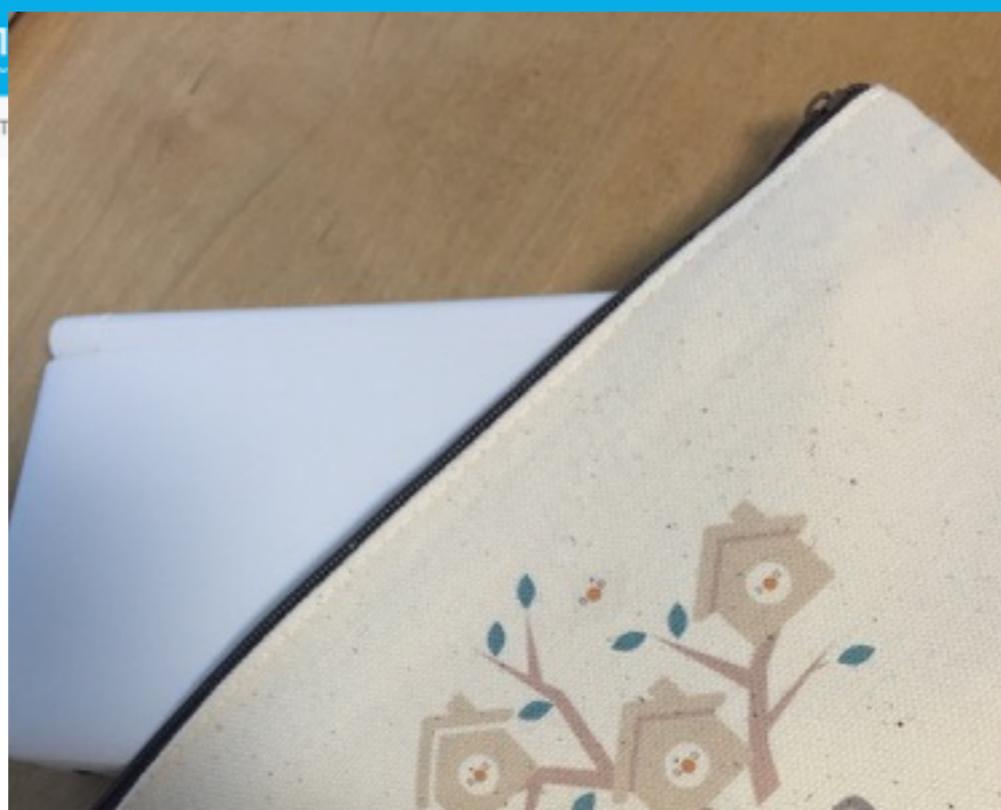
デザインID : 2227905  
0Θ0



このデザインを購入する



オリジナルTシャツ HOME > esa 0Θ0 和 ショク ミート のデザイン





**tmix.jp**

**自分たちの  
ために作る**



**steers.jp**

**販売する**



在庫不要・送料コミ  
利用料・制作費 **0円**

ウラ・オモテ  
両面プリントOK!

フルカラー

**3枚**の  
オーダーで  
販売確定  
▶詳しくはこちら

だれでもかんたんに作って売れる  
オリジナルTシャツ販売の STEERS (ステアーズ)



## STEERSでは誰でも簡単Tシャツのデザイン&販売ができます



Tシャツをデザインして  
企画を立ち上げる

STEERSでは、だれでも簡単にオリジナルTシャツのデザインと販売ができます



みんなに知らせて  
購入者をつのる

作ったTシャツをTwitterやFacebookで広めて、購入者をつのりましょう



梱包・配送は  
すべてお任せ

売れたTシャツの配送はまるごとおまかせ！あとは利益を受け取るだけです

## 販売コスト・在庫リスクはゼロ！利益は自由に設定できます。

たとえば、白地に片面インク1色デザインのTシャツを  
1枚 2,800 円で販売したら利益はこんな感じ

5枚 売れたら - - - - - 5,250円

10枚 売れたら - - - - - 10,500円

50枚 売れたら - - - - - 52,500円

100枚 売れたら - - - - - 105,000円

お試し販売もお気軽に

売れても売れなくても  
システム利用料は  
完全無料で安心

STEERSサポートデスク  
E-mail: support@steers.jp  
フリーダイヤル: 0120-910-157  
(受付時間 平日 9:00 - 18:00)



Fine Fit T-shirts  
DM501

なめらかな質感と  
適度に細身で美しい  
シルエットが自慢！



コーミングにより長い繊維だけが平行に揃えられた綿100%のコーマ糸を使用。  
毛羽立ちが少なくなめらかな仕上がり。糸の目を詰まらせることで、型崩れしにくく  
いしっかき感のあるクオリティを実現。  
広めの襟あきが美しいレディースサイズもご用意しています。

### MATERIAL

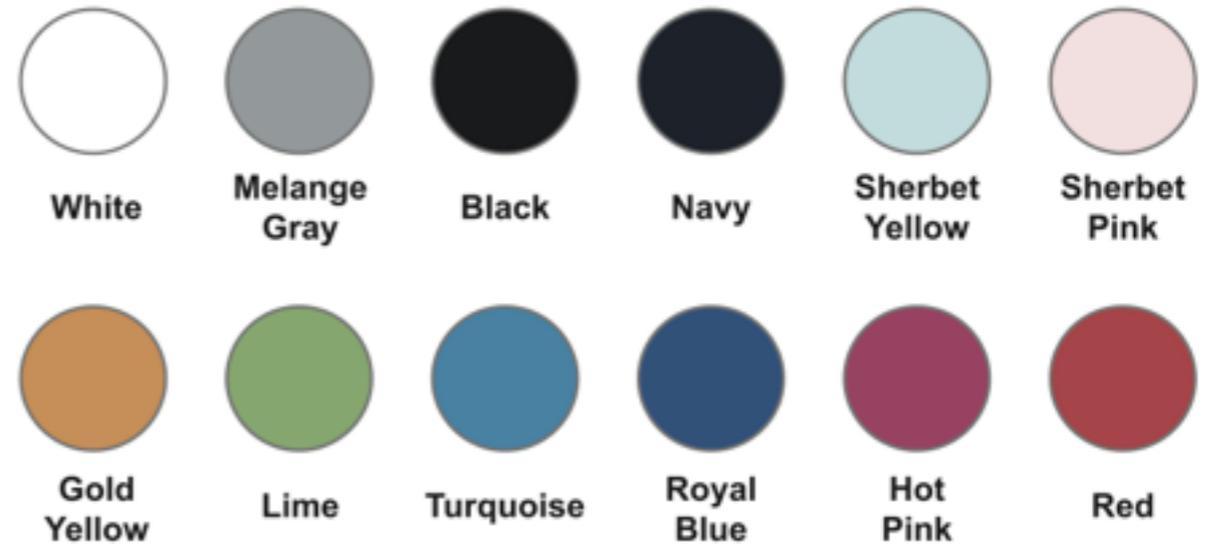
- 4.6oz
- 天竺 コーマ糸
- コットン 100% (メランジェグレーのみレーヨン混)

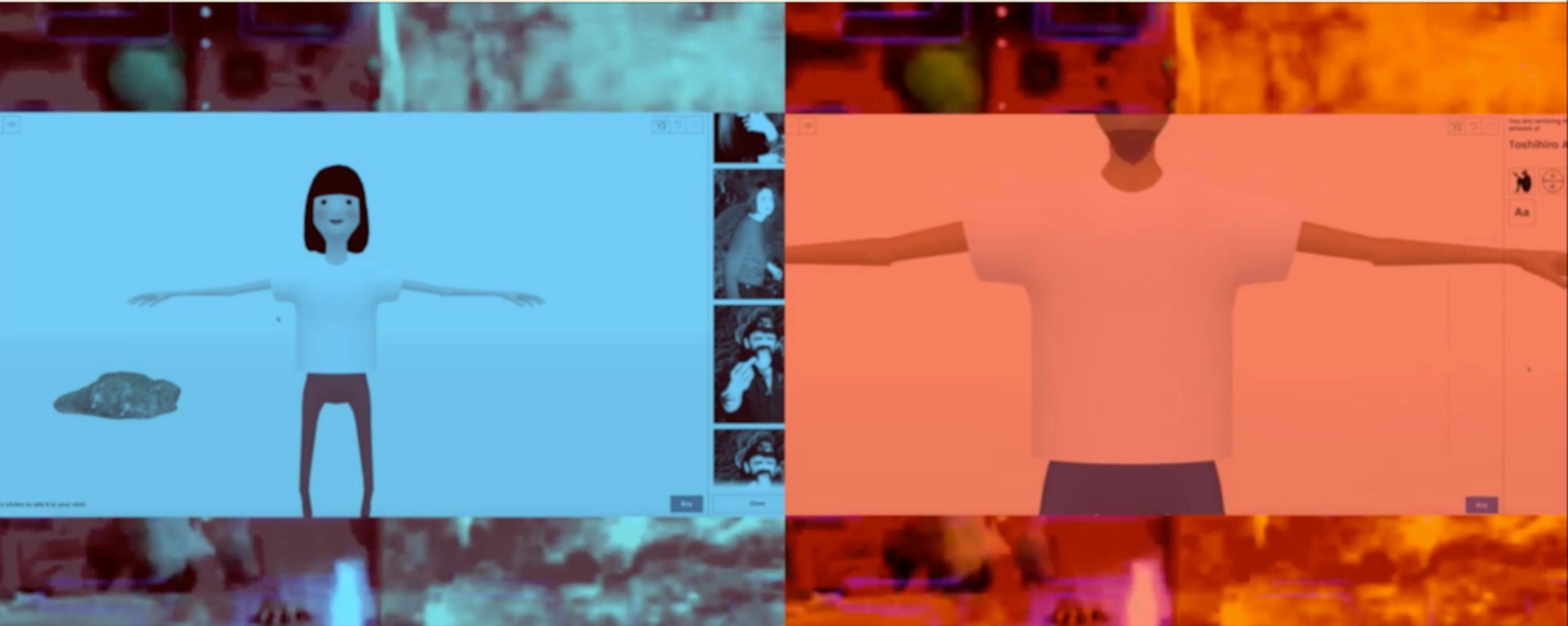


### SIZE

- レディースサイズ - WM~WL
- 男女兼用サイズ - XS~XL

### COLOR





to create something beautiful and new. Collaborate with a group of amazing artists from NYC and Tokyo! Blow up and warp their work ho

# DCF : Disco Curly Fries

with to.be



# 株式会社 spice life デザイナー、エンジニア募集中!!



ポケストップ、ジム完備

試験公開中



# 達人出版会

Home

Books



## Railsの教科書

五十嵐邦明

[達人出版会](#)

700円 490円+税 《特価》 [PDF](#) [EPUB](#)

**β版** プログラミング初学者へ向けたRailsを使ったアプリを題材に、その動作と仕組みを解説し、他の書の基礎の部分を丁寧に説明していきます。基礎力を

買い物かごへ

ギフトで購入



# Railsの講義

# Railsの教科書

## 方針

- 対象はプログラミングが初めての方
- できるだけ分かり易く説明したい

# Railsの教科書

## Webアプリの基本動作 - レスポンス

レスポンスとしてHTMLが返ってくる

リクエスト

URL : http://cookpad.com/



## 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

## パラメータの中身は？

ここで出力した params の値と、さきほどブラウザで表示させたパラメータの値が同じことが分かります。ブラウザがユーザーの入力データをパラメータとして送信し、私たちが作成しているアプリがそれを受け取っていることが確認できました。 **ブラウザのデベロッパーズツール**



コンソールのアプリログ

```
{:utf8=>"/", :authenticity_token=>"zGLyQpUxcQA15Ca1AT50b8Nxr3oU7DUZqH57Nq7Fml", :book=>{:title=>"大塚反トイボックス", :memo=>"ゲーム開発会社を舞台にした動物ストーリー"}, :commit=>"create book", :action=>"create", :controller=>"b"}
```

## 今回つくったRailsアプリの動作

リクエスト

- ▶ URL : http://localhost:3000/hello/index



## Railsアプリの基本的なフォルダ・ファイル群

rails new helloworld を実行すると、以下のフォルダ・ファイルが生成されます。(一部のみ抜粋)

- helloworld ← アプリ名のフォルダ
  - app
    - controllers ← いろいろな処理を行う箇所
    - views ← 表示に関する処理を行う箇所
  - config
    - routes.rb ← URLとコントローラの対応表
  - ...

## 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、特に指定がない場合はコントローラ名/アクション名的なViewへ移動します。→次はViewである app/views/hello/index.html.erb に処理が渡ります。

# 実際に作って何ができるかを知る

## 題材: 画像投稿機能付きミニブログアプリ

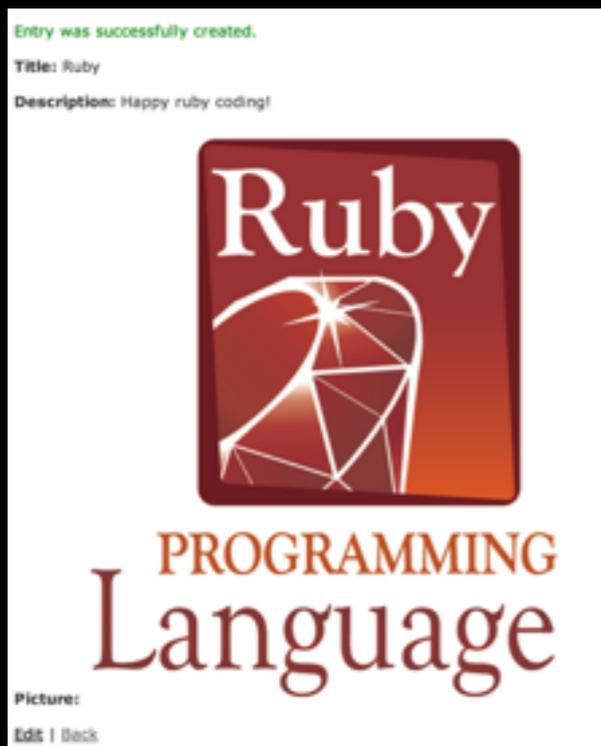
### New entry

Title

Description

Picture  
ファイルを選択 選択されていません

[Back](#)



## Railsの教科書より一部抜粋

```
rails new blog_app
```

以下のような結果が表示されるでしょうか。(実行には少し時間がかかります)

```
$ rails new blog_app
```

```
create
create README.rdoc
create Rakefile
```

```
... (略)
```

```
Your bundle is complete!
```

```
Use `bundle show [gemname]` to see where a bundled gem is installed.
```

```
run bundle exec spring binstub --all
```

```
* bin/rake: spring inserted
```

```
* bin/rails: spring inserted
```

ターミナルの画面にこのように"Your bundle is complete!"と表示されれば成功です。railsコマンドはたくさんのフォルダとファイルを自動で作ります。

次は以下のコマンドを実行してみてください。

```
cd blog_app
```

```
rails server
```

うまく動作している場合は、以下のような表示になります。

```
$ rails server
```

```
=> Booting WEBrick
```

```
=> Rails 4.1.0 application starting in development on http://0.0.0.0:3000
```

```
=> Run `rails server -h` for more startup options
```

# サンプルアプリで手本となるコードに触れる 画像投稿機能付きミニブログアプリ

## 学ぶ要素

- **scaffold** (生成されたコードは良い手本)
  - **CRUD基礎**
  - **MVC**
  - **Railsの処理の流れ**
    - **routes, controller, view**
- **Gem, Bundler** (画像投稿: `carrierwave` gem)

試験公開中



# 達人出版会

Home

Books



## Railsの教科書

五十嵐邦明

[達人出版会](#)

~~700円~~ 490円+税 《特価》 [PDF](#) [EPUB](#)

**β版** プログラミング初学者へ向けたRailsを使ったアプリを題材に、その動作と仕組みを解説し、他の書の基礎の部分を丁寧に説明していきます。基礎力を

買い物かごへ

ギフトで購入

# Railsの教科書 top

[はじめに](#)

[Webアプリをまずはつくってみよう](#)

[一番小さなRailsアプリづくり](#)

[CRUDの基礎とindexアクション](#)

[new, create アクション](#)

[モデル](#)

**<http://bit.ly/igazemi>**

---

**[https://github.com/igaiga/rails\\_textbook](https://github.com/igaiga/rails_textbook)**

# Railsをもっと学びたい人へ

## Web上の資料

- **Railsの教科書** <http://bit.ly/igazemi>
- **Rubyの教科書** <https://github.com/igaiga/hitotsubashi-ruby-2013>
- **RailsGuides** <http://railsguides.jp/>
  - 網羅的な解説でRailsの機能を学ぶ
  - 達人出版会に電子書籍版もあり <http://tatsu-zine.com/books/railsguides>
- **RailsTutorial** <http://railstutorial.jp/>
  - サンプルアプリづくりを通じてRailsを学ぶ
  - 達人出版会に電子書籍版もあり <http://tatsu-zine.com/books/railstutorial>



# 株式会社 spice life



**デザイナー、エンジニア募集中!!**





**T MIX**



***STEEERS***



# http://bit.ly/tmix-rgk03a



0120-570-150  
月~土 9:00~18:00(祝日除く)



いがいがfbさん

[アイテムと価格](#)

[デザイン](#)

[法人・物販](#)

[TMIXについて](#)

[ショールーム](#)

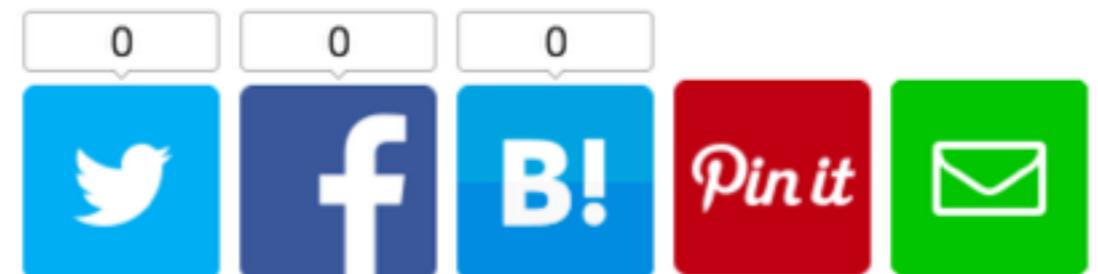
[新着情報](#)

[オリジナルTシャツ HOME](#) > [RailsGirlsKyoto3のデザイン](#)

## RailsGirlsKyoto3

デザインID : 2092736

[myページ](#)からコメントを編集できます



▶ [このデザインを購入する](#)

# http://bit.ly/tmix-esa06



0120-570-150  
月～土 9:00～18:00 (祝日除く)



アイテムと価格

デザイン

法人・物販

TMIXについて

ショールーム

新着情報

オリジナルTシャツ HOME > esa 0⊖0 和 ショク ミート のデザイン

## esa 0⊖0 和 ショク ミート

デザインID : 2227905

0⊖0



▶ このデザインを購入する

# http://bit.ly/tmix-esa07



0120-570-150

月～土 9:00～18:00 (祝日除く)



カート



いがいがfbさん

[アイテムと価格](#)

[デザイン](#)

[法人・物販](#)

[TMIXについて](#)

[ショールーム](#)

[新着情報](#)

[オリジナルTシャツ HOME](#) > [esaシャクパ°-カー](#)のデザイン

## esaシャクパ°-カー

デザインID : 2245883

[myページ](#)からコメントを編集できます



▶ [このデザインを購入する](#)

<http://bit.ly/tmix-esa05>



# Railsの教科書 学習の流れ

- 実際に作って何ができるかを知る
- Railsアプリの仕組みを知る
- 手本となるコードに触れる
- デバッグの方法を知る
- 課題をこなしていく

# Railsアプリの仕組みを知る

## Webアプリの基本動作 - レスポンス

レスポンスとしてHTMLが返ってくる

リクエスト

URL : http://cookpad.com/



## 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

## パラメータの中身は？

ここで出力した params の値と、さきほどブラウザで表示させたパラメータの値が同じことが分かります。ブラウザがユーザーの入力データをパラメータとして送信し、私たちが作成しているアプリがそれを受け取っていることが確認できました。 **ブラウザのデベロッパーツール**



コンソールのアプリログ

```
{:utf8=>"", :authenticity_token=>"zGLyQpUxcQA15Ca1AT50b8Nxr3oU7DUZqH57Nq7Fml", :book=>{:title=>"大塚反トイボックス", :memo=>"ゲーム開発会社を舞台にした動物ストーリー"}, :commit=>"create book", :action=>"create", :controller=>"b"}
```

## 今回つくったRailsアプリの動作

リクエスト

- ▶ URL : http://localhost:3000/hello/index



## Railsアプリの基本的なフォルダ・ファイル群

rails new helloworld を実行すると、以下のフォルダ・ファイルが生成されます。(一部のみ抜粋)

- helloworld ← アプリ名のフォルダ
  - app
    - controllers ← いろいろな処理を行う箇所
    - views ← 表示に関する処理を行う箇所
  - config
    - routes.rb ← URLとコントローラの対応表
  - ...

## 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、特に指定がない場合はコントローラ名/アクション名的なViewへ移動します。→次はViewである `app/views/hello/index.html.erb` に処理が渡ります。

Railsの教科書より一部抜粋

# デバッグの方法を知る



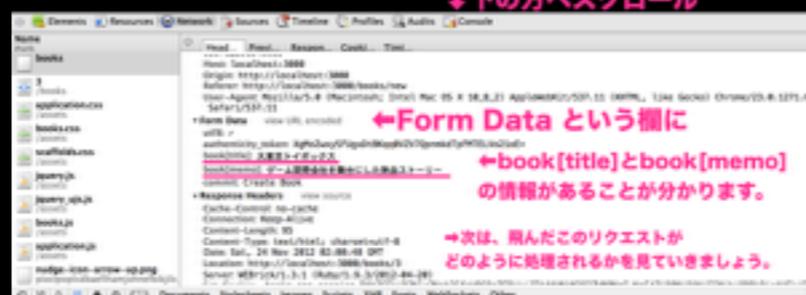
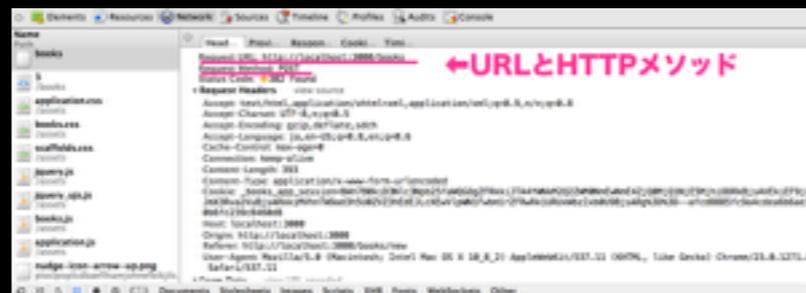
- エラーの読み方
- ログの出し方、読み方
- ブラウザのデバッグ機能
- pry でデバッグ

<https://speakerdeck.com/igaiga/rubyconftaiwan2015>

## ブラウザに表示されるエラーを読む



エラー例：  
app/controllers/  
books\_controller.rb  
の5行目で bar っていう知らないものがあるのでエラーだよ。



## パラメータの中身は？

app/controllers/books\_controller.rb

```
def book_params
  p *****
  p params
  params.require(:book).permit(:title, :memo)
end
```

実行結果を見ると、確かに paramsの中にブラウザで入力した値が入っていることがわかりました。(形式はHashですね。)

```
Started POST "/books" for 127.0.0.1 at 2013-09-23 06:47:19 +0900
Processing by BooksController#create as HTML
Parameters: {"utf8"=>"✓", "authenticity_token"=>"zGlyQpUxcQA1SCaLAT50b0Nxr3oU7DUZgH57NmQ77mE=", "book"=>{"title"=>"大東京トイボックス", "memo"=>"ゲーム開発会社を舞台にした熱血ストーリー"}, "commit"=>"Create Book"}
*****
{"utf8"=>"✓", "authenticity_token"=>"zGlyQpUxcQA1SCaLAT50b0Nxr3oU7DUZgH57NmQ77mE=", "book"=>{"title"=>"大東京トイボックス", "memo"=>"ゲーム開発会社を舞台にした熱血ストーリー"}, "commit"=>"Create Book", "action"=>"create", "controller"=>"books"}
(0.1ms) begin transaction
```