

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 1 章 この授業について

## 1 導入

## 2 学習目的・目標

## 3 授業の方法

## 4 演習課題

# はじめに

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## 自己紹介

- `enpit/self_introduction.org` at master · ychubachi/enpit

## 資料の入手

- 資料は GitHub からダウンロードできる `ychubachi/framework_development`
- 随時更新するので watch しておくとよい

# 授業の計画

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## スケジュールの調整

- 基本的には資料の通り進めるが、計画の変更はあり得る

## 休講・補講

- なし（予定）

## 1 導入

## 2 学習目的・目標

## 3 授業の方法

## 4 演習課題

# この授業の目的

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- 再利用可能なコンポーネント開発の概念的理解を行う
- 基本技術を押さえて、高度なフレームワークを使いこなせるようになる
  - Ruby の Gem を題材に、実装技術を学ぶ

# この授業の目標

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- Ruby で実際にコーディングを行い，コンポーネントを開発できる
- Git/GitHub を用い，開発の効率の向上・成果物の公開などができる
- Ruby のテスト技術について学ぶ
- RubyGems に公開する方法



## 1 導入

## 2 学習目的・目標

## 3 授業の方法

## 4 演習課題

# この授業で用いる資料

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- この授業で取り上げる資料は、開発コミュニティが公開している Web ページを中心に解説する
- エンジニアは英語の原典を読めなくてはならないので、英語のページを見ながら解説する

# Git/GitHub の活用

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- GitHub のアカウントを作成しておくこと
- ソースコードを作成する課題は，GitHub にも登録してもらうことがある

# 仮想化環境の準備

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- Ruby の開発環境
  - enPiT の仮想環境を利用
- 目標
  - 「Vagrant」を利用できるようにする
- バージョン
  - VirtualBox 4.3.28
  - Vagrant 1.7.2

# インストール方法（Windows）

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## ■ 資料

- `enpit/preparation.org` at master · ychubachi/enpit

## ■ 動画を参照

- <http://youtu.be/kePqg8dCgJM>

# インストール方法 (Mac)

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## Homebrew のインストール

- Homebrew —The missing package manager for OS X
- Homebrew Cask

## VB と Vagrant のインストール

```
brew cask install virtualbox  
brew cask install vagrant
```

# 仮想化環境の設定

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## 内容

- enPiT 用に作成した Vagrant の box ファイルを入手して実行できるようにする
- Vagrant のバージョンは最新版にしておく
- Vagrant のインストール後、次のコマンドでインストール可能

## コマンド

```
vagrant init ychubachi/enpit
```

# 仮想環境利用時の注意事項

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

- 仮想環境は簡単に削除できる
  - 保存したいファイルは必ずクラウド（GitHub）や、Host 側のファイルシステムに残すこと
- 仮想環境の更新
  - ychubachi/enpit は授業期間中でもバージョンアップする可能性がある
  - この場合も、Guest 側のファイルは消えるので注意
- ツールのバージョン
  - vagrant, VirtualBox を以前にインストールしたことのある場合、最新のバージョンに更新しておくこと



## 1 導入

## 2 学習目的・目標

## 3 授業の方法

## 4 演習課題

# 課題 1-1

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## 開発環境の構築

- 演習用仮想化環境を用意する
- 資料と動画を参照

## 作業内容

- VirtualBox と Vagrant をインストールする
  - Oracle VM VirtualBox
  - Vagrant

# 課題 1-2

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## 開発環境の構築

- enPiT 仮想化環境をインストールする

## コマンド

```
vagrant init ychubachi/enpit  
vagrant up  
vagrant ssh
```

# 課題 1-3（上級者向け）

フレームワーク開発特論  
(2015)

中鉢欣秀

導入

学習目的  
・目標

授業の方法

演習課題

## 仮想環境自体の構築

- enPiT 用の Vagarnt の Box 自体を作成するスクリプト等は次のリポジトリにある
  - [https://github.com/ychubachi/vagrant\\_enpit](https://github.com/ychubachi/vagrant_enpit)

## 内容のレビュー

- Chef 等をインストールして Box を作成してみよ
- 突っ込み所は多いと思われるので、改善点に気がいたら issue にあげよ
- Pull request も歓迎する

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 2 章 Ruby 自体に関すること

フレームワー  
ク開発特論  
(2015)

中鉢欣秀

Ruby のバージ  
ョン指定

動的な言語と  
しての Ruby

演習課題

## 5 Ruby のバージョン指定

## 6 動的な言語としての Ruby

## 7 演習課題

# Ruby のバージョン

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- Ruby には様々なバージョンがある
  - 最新の安定版: Ruby 2.2.2
  - 1.8.7, 1.9.2 はサポート終了
- 日々激しくバージョンアップされる
- これら以外にも, Java や .NET Framework 上で動作するものなど多数.

# 開発時の混乱

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- 開発プロジェクトによって、異なるバージョンの Ruby が用いられる
- 複数の開発プロジェクトに参加する開発者が、毎回手動でバージョンを変更するのは困難であるし、トラブルの原因となる



# Rbenv について

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

## 概要

- バージョンの異なる複数の Ruby を管理するツールであり、主要なエコシステムの一部
- プラグインを追加することで、インストールも自動化できる

## GitHub

- [sstephenson/rbenv](#)

## RVM との比較

- [Why rbenv?](#) [sstephenson/rbenv Wiki](#)

フレームワー  
ク開発特論  
(2015)

中鉢欣秀

Ruby のバージ  
ョン指定

動的な言語と  
しての Ruby

演習課題

## 5 Ruby のバージョン指定

## 6 動的な言語としての Ruby

## 7 演習課題

# Ruby の REPL

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

## irb とは

- irb = Interactive Ruby
  - REPL
- Ruby のコードが実行できる
  - 構文や動作の確認に便利

## 起動と終了

- 起動するには irb
- 終了するには exit (または Ctrl-d)

# Ruby は動的な言語

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

## 機能の追加の例

```
require 'pp'  
# pp ENV
```

## 機能の置換の例

```
require 'mathn'  
# 10/3
```

## 再起動すると？

- 拡張は初期化される

# 参考文献

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- Ruby にはウンザリ！動的型付け、副作用、およびオブジェクト指向プログラミング全般からの考察 | プログラミング | POSTD

# ロードパス

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

## `$LOAD_PATH`

- 既存のライブラリや別ファイルを読み込む際、検索の対象となる `PATH`

## `$LOADED_FEATURES`

- 一度 `require` されたファイルは `$LOADED_FEATURES` に追加される

## 5 Ruby のバージョン指定

## 6 動的な言語としての Ruby

## 7 演習課題

## 課題 2-1 別な Ruby バージョンのインストール

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- `rbenv` を用いて、異なるバージョンの Ruby をインストールする



## 課題 2-2 Ruby のバージョンを切り替える

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- rbenv を使い, ruby のバージョンを切り替える

## 課題 2-3 irb を使ってみよう

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- irb を用い，ruby の様々な構文を試しなさい

## 課題 2-4 標準ライブラリ

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- 次のような標準ライブラリを使用してみなさい
  - yaml
  - set
  - net/http
  - erb

## 課題 2-5 （上級） pry について

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby のバージョン指定

動的な言語としての Ruby

演習課題

- irb と似たツールに pry がある
- これを使用して見て irb との差異を調べなさい

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 3 章 RubyGems の概要と周辺のツール群

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

## 8 Ruby の Gem とは

## 9 RubyGems の解説 (1)

## 10 RubyGems の解説 (2)

## 11 補足

## 12 演習課題

# Ruby によるコンポーネント

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- Ruby には、再利用可能なコンポーネントを取り扱う仕組みとして Gem がある
- 資料
  - RubyGems Guides

# 基本的なコマンド

- gem コマンドは，Ruby でコンポーネントを開発したり，配布をしたりする等の際に利用するコマンド
- 後に解説する bundler コマンドの基盤
- 資料
  - RubyGems Basics - RubyGems Guides

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題



フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

8 Ruby の Gem とは

9 RubyGems の解説 (1)

10 RubyGems の解説 (2)

11 補足

12 演習課題

# Ruby Gems のガイド

フレームワーク  
開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- What is a gem? - RubyGems Guides
- Make your own gem - RubyGems Guides
- Gems with Extensions - RubyGems Guides
  - C 言語拡張 (省略)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

8 Ruby の Gem とは

9 RubyGems の解説 (1)

10 RubyGems の解説 (2)

11 補足

12 演習課題

# Ruby Gems のガイド

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- Name your gem - RubyGems Guides
- Publishing your gem - RubyGems Guides
- Security - RubyGems Guides
  - セキュリティ (省略)
- Patterns - RubyGems Guides

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

8 Ruby の Gem とは

9 RubyGems の解説 (1)

10 RubyGems の解説 (2)

11 補足

12 演習課題

# MiniTest

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

■ library minitest/unit

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

8 Ruby の Gem とは

9 RubyGems の解説 (1)

10 RubyGems の解説 (2)

11 補足

12 演習課題

# 課題 3-1

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- 下記のガイドに記されたサンプルを実行し, `gem` コマンドの基本的な使い方学ぶ
- RubyGems Basics - RubyGems Guides



## 課題 3-2

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- gem をダウンロードして中身を見してみる
- search, fetch, unpack などのコマンドを活用する

## 課題 3-3 RubyGems.org にアカウントを作成

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- RubyGems.org にアカウントを作成しなさい
- RubyGems.org | your community gem host
  - 「sign up」リンクから作成する

## 課題 3-4 ガイドを参考に Gem を作る

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby の Gem  
とは

RubyGems の  
解説 (1)

RubyGems の  
解説 (2)

補足

演習課題

- ガイドの解説に従い, "hola" Gem を作成しなさい
  - Make your own gem - RubyGems Guides
- 演習用 Gem の名前の付け方
  - hola\_(username)
  - 括弧内は RubyGems のユーザ名に置き換えよ

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 4 章プロジェクトで利用する Gem の管理

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

## 13 プロジェクトと Gem

## 14 Bundler で Gem を利用する

## 15 Bundler による Gem の作成

## 16 TODO 補足資料

## 17 演習課題

# プロジェクトごとに異なる Gem の集合

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

- 開発プロジェクトにおいて利用する Gem は異なるばかりではなく、Gem のバージョンについても注意が必要である
- Gem コマンドを直接用いてインストールする方法では、必要な Gem を主導で管理しなくてはならないし、バージョンの異なる Gem を用いる場合もある

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

13 プロジェクトと Gem

14 Bundler で Gem を利用する

15 Bundler による Gem の作成

16 TODO 補足資料

17 演習課題

# Bundler について

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

- Bundler: The best way to manage a Ruby application's gems



# ドキュメントの解説

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

- Why Bundler exists
- Understanding binstubs · sstephenson/rbenv Wiki
- Gemfile

# bundle コマンドのインストール

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

## bundle コマンド

- gem と違い, bundle コマンドは標準ではインストールされていない.

## Gem によるインストール方法

```
gem install bundler
```

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

13 プロジェクトと Gem

14 Bundler で Gem を利用する

15 Bundler による Gem の作成

16 TODO 補足資料

17 演習課題

# ひな形の自動生成

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

## ひな形の自動生成

- Bundler: The best way to manage a Ruby application's gems

## コマンド

```
bundle gem my_gem_name
```

# 雛形の内容

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

```
.
—— .git
    <snip>
—— .gitignore
—— Gemfile
—— LICENSE.txt
—— README.md
—— Rakefile
—— lib
    —— my_gem_name
        —— version.rb
    —— my_gem_name.rb
—— my_gem_name.gemspec
```

# git ls-files について

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

- git に index されているファイルの一覧
- 新しいファイルは，ステージング領域に add されると表示に加わる
- .gitignore で無視するファイルを設定できる

# 雛形の build

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

- \*.gemspec の TODO を外す（内容をきちんと書く）
- rake build で build できる

# executable の作成と注意

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

- `bundle gem -b` で実行可能なスクリプトの雛形ができる
- 実行属性を `chmod a+x` でつける
- `bundle install --binstubs` を実行すると、上書きされるので注意
  - 手動で他のディレクトリ（`/exe`）に作成するほうが良い



# Gemfile と gemspec の関係

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

## ■ Gem の依存関係のかき分け

### ■ Gemfile

- Gemfile に gemspec メソッドがあれば, .gemspec 内の Gem 依存関係を解決する

### ■ \*.gemspec

- Gem を利用時に必要な Gem を追加

# Bundler と \$LOAD\_PATH

フレームワーク  
開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

## Bundler と \$LOAD\_PATH

- Bundler はロードパスを適切に設定

## ロードパスの設定の確認

- `bundle exec irb`

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

**TODO** 補  
足資料

演習課題

13 プロジェクトと Gem

14 Bundler で Gem を利用する

15 Bundler による Gem の作成

16 **TODO** 補足資料

17 演習課題

# 補足資料

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

## GitHub への公開鍵登録など

- `github-connect.sh`
- 注意：画面が乱れたら再ログイン

## GitHub 用 CLI

- 演習用仮想環境には `hub` コマンドがインストール済み `github/hub`

## Editor

- VIM にする方法 `git config --global core.editor "vim"`

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

13 プロジェクトと Gem

14 Bundler で Gem を利用する

15 Bundler による Gem の作成

16 TODO 補足資料

17 演習課題

## 4-1 Bundler で簡単な Gem を作ってみる

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利用する

Bundler による  
Gem の作成

TODO 補  
足資料

演習課題

### 課題

- 簡単な計算を行うコマンドを Gem として作成しなさい.
- 外部の Gem を利用する場合は, gemspec ファイルに依存関係を記述すること

### 例

- 生年月日と今の年月日から年齢を計算する
- 身長と体重を入力して, BMI を出す
- その他, 各自で考えよ

## 4-2 作成した Gem を GitHub で公開する

フレームワーク  
開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

### 課題

- 作成した Gem のソースコードを GitHub に公開しなさい
  - `hub create` で GitHub のリポジトリを作成
  - `git add`, `git commit` でコミット
  - `git push -u origin master` で GitHub に登録

### 提出

- GitHub の URL を LMS に提出する

## 4-3 Gem を RubyOrg に登録する

フレームワーク開発特論  
(2015)

中鉢欣秀

プロジェクト  
と Gem

Bundler で  
Gem を利  
用する

Bundler によ  
る Gem の作成

TODO 補  
足資料

演習課題

### 課題

- 作成した Gem を RubyOrg に登録する
- Gem の名前には, aiit のアカウント名を先頭につけること
  - a14???xx\_name

### 提出

- RubyOrg の URL



# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 5 章 Rake によるタスクの実行

## 18 Rake の文書

## 19 簡単な Rakefile の例

# ドキュメント

フレームワー  
ク開発特論  
(2015)

中鉢欣秀

Rake の文書

簡単な  
Rakefile の例

- [jimweirich/rake](#)
- [Rake Part 1: Files and Rules | Virtuous Code](#)
- [Nilquebe Blog: Rake Part 1: Files and Rules](#) 翻訳

## 18 Rake の文書

## 19 簡単な Rakefile の例

# タスクの定義

フレームワーク開発特論  
(2015)

中鉢欣秀

Rake の文書

簡単な  
Rakefile の例

## Rakefile

```
task :hello do
  puts 'do task hello!'
end
```

# 説明を追加した例

フレームワーク開発特論  
(2015)

中鉢欣秀

Rake の文書

簡単な  
Rakefile の例

## Rakefile

```
desc 'say hello'
task :hello do
  puts 'do task hello!'
end
```

# タスクの一覧

フレームワーク  
開発特論  
(2015)

中鉢欣秀

Rake の文書

簡単な  
Rakefile の例

## コマンド

```
rake -T
```

# Bundler が自動生成する gem

フレームワーク  
開発特論  
(2015)

中鉢欣秀

Rake の文書

簡単な  
Rakefile の例

## Rakefile

```
require "bundler/gem_tasks"
```

## タスクの定義

- 実際のタスクは `bundler/gem_tasks` 内にある
- なお、コードの在処は `gem which bundler` で確認できる



# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 6 章 Ruby による単体テスト

## 20 Ruby によるテスト技法

## 21 テストを含む Gem の生成

## 22 minitest の書き方

## 23 演習課題

# 各種のテスト技法 (1)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitestの書き方

演習課題

## ■ Minitest

- Ruby に標準のテストツール
- Module: MiniTest (Ruby 2.0.0)

## ■ RSpec

- 広く普及しているテストツール
- 「振る舞い駆動」
- [RSpec.info: home](http://RSpec.info)

# 各種のテスト技法 (2)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

- Cucumber
  - Cucumber - Making BDD fun
- Turnip
  - jnicklas/turnip

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含む  
Gem の生成

minitest の  
書き方

演習課題

20 Ruby によるテスト技法

21 テストを含む Gem の生成

22 minitest の書き方

23 演習課題

# bundle gem のオプション

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含む  
Gem の生成

minitest の  
書き方

演習課題

## minitest を使う場合

```
bundle gem gem_minitest --test=minitest
```

# minitest 用の Rakefile

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含む Gem の生成

minitest の書き方

演習課題

## Rakefile

```
require "bundler/gem_tasks"
require "rake/testtask"

Rake::TestTask.new(:test) do |t|
  t.libs << "test"
end

task :default => :test
```

# test/の中身

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

## ファイル

```
test
|-- minitest_helper.rb
`-- test_gem_minitest.rb
```

## 内容

`minitest_helper.rb` テストを実行する際に必ず読み込まれる

`test_gem_minitest.rb` テストを書く場所



## 20 Ruby によるテスト技法

## 21 テストを含む Gem の生成

## 22 minitest の書き方

## 23 演習課題

# minitest のドキュメント

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

## ドキュメント

- 英語版
  - Module: MiniTest (Ruby 2.1.0)
- 日本語版
  - library minitest/unit

# Assertion

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

## ■ Assertion とは？

■ `module MiniTest::Assertions`

## 20 Ruby によるテスト技法

## 21 テストを含む Gem の生成

## 22 minitest の書き方

## 23 演習課題

## 6-1 Gem の作成

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテ  
スト技法

テストを含む  
Gem の生成

minitest の  
書き方

演習課題

### 課題

- テストの演習をするための Gem を作成しなさい

### コマンド

```
bundle gem minitest_practice --test=minitest
```

### サンプルコード

- `ychubachi/minitest_sample`

## 6-2 メソッドの作成 (1)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば、コードを書きなさい

### 仕様

メソッド名 odd?

内容 整数を入力として受け取り、値が奇数ならば真を返す

## 6-3 メソッドの作成 (2)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば、コードを書きなさい

### 仕様

メソッド名 `check_number?`

内容 引数が 0 以外ではじまる 4 桁の数字であり，なおかつ，値が偶数ならば真を返す

## 6-4 メソッドの作成 (3)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば，コードを書きなさい

### 仕様

**メソッド名** `enough_length?`

**内容** 文字列を受け取り，その長さが 3 文字以上，8 文字以下であれば真を返す



## 6-5 メソッドの作成 (4)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば、コードを書きなさい

### 仕様

メソッド名 divide

内容 引数として割る数と割られる数を取り、割り算をした結果を返す。  
ただし、0 で割り算をしたら例外を発生する

## 6-6 メソッドの作成 (5)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば、コードを書きなさい

### 仕様

メソッド名 `fizz_buzz`

内容 引数に数値を1つとる. 3の倍数の時は”Fizz”を返す. 5の倍数の時は”Buzz”を返す. 3と5の公倍数のときは”FizzBuzz”を返す.

## 6-7 メソッドの作成 (6)

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

### 課題

- 次の仕様に沿ったテストを作成しなさい
- テストができれば、コードを書きなさい

### 仕様

- 標準出力に「Hello」と表示するプログラムのテストと実装を行いなさい

# 参考

フレームワーク開発特論  
(2015)

中鉢欣秀

Ruby によるテスト技法

テストを含むGemの生成

minitest の書き方

演習課題

## サンプル

■ [https://github.com/ychubachi/mini\\_test\\_practice](https://github.com/ychubachi/mini_test_practice)

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第7章 テスト自動化と統合テスト

## 24 Guard による方法

## 25 Travis CI による方法

## 26 演習課題

# Guard とは？

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

- Guard とは，ファイルの更新を監視して，更新があれば指定されたタスクを実行する仕組み
- 詳細
  - `guard/guard`

# MiniTest の自動化

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

- Guard にはプラグイン機能がある
- MiniTest 用のプラグイン
  - guard/guard-minitest
- Gemfile に追加し，bundle install が必要



# サンプル

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

## ■ Guard 用の設定を行ったコード

■ `ychubachi/mini_test_practice`

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による  
方法

Travis CI による  
方法

演習課題

24 Guard による方法

25 Travis CI による方法

26 演習課題

# Travis CI とは？

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

- GitHub と連携し、新たなコミットが GitHub に Push されたら自動でテスト（など）を行う機能
- CI (continuous integration)
  - 継続的統合などと呼ばれる

# Travis CI について

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

- <https://travis-ci.org/>
- サンプルを実行する例
- 設定は `.travis.yml` に書く

# GitHub と連携する方法

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

## ■ 設定ファイル

- `.travis.yml` は `bundle gem` コマンドを実行した段階で生成されている

## ■ GitHub のフックを設定する

- 1 GitHub でプロジェクトのリポジトリを開く
- 2 Settings -> Webhooks & Services
- 3 Add Services ボタンから Travis CI を選択

# コマンドで行う場合

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

- `travis` コマンドをインストール
- `travis enable` コマンドで連携開始

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

24 Guard による方法

25 Travis CI による方法

26 演習課題

# 7-1 Guard

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

## 課題

- Guard を利用して、テストを自動化しなさい



## 7-2 Travis CI

フレームワーク開発特論  
(2015)

中鉢欣秀

Guard による方法

Travis CI による方法

演習課題

### 課題

- GitHub と Travis CI を連携させ、継続的統合を行いなさい

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

## 第 8 章 GitHub を用いた開発演習

## 27 GitHub とは

## 28 ペアで行う GitHub

## 29 グループで行う GitHub

## 30 TODO 補足資料

# GitHub について

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

## ■ Web サイト

- GitHub
- GitHub Japan

## ■ 主な機能

- ソースコード共有 (Git)
- Issue
- Pull requests
- Wiki
- コード解析

# GitHub Flow

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

- Git-flow
  - GitHub が登場する以前、Git-flow が提唱された
  - A successful Git branching model » [nvie.com](https://nvie.com)
- GitHub flow
  - GitHub により、よりシンプルで強力なワークフローが可能に
  - GitHub Flow – Scott Chacon
  - GitHub Flow (Japanese translation)
- この演習では GitHub flow を学ぶ

# GitHub flow におけるコンフリクトについて

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行う  
GitHub

TODO 補  
足資料

## ■ マージのコンフリクト

- GitHub に提出した Pull requests が自動的にマージできないこと

## ■ 基本的な対処法

- コンフリクトは、コードの同じ箇所を複数の人が別々に編集すると発生
- 初心者は、演習の最初の方では「他人と同じファイルを編集しない」ことにして、操作になれる
- 上達したら積極的にコンフリクトを起こしてみて、その解決方法を学ぶ
- Pull requests でコンフリクトが発生し、自動的にマージできない状態になったら、その PR を送った人がコンフリクトを自分で解消する

# リポジトリの作成と clone

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

## リポジトリの作成方法

- git create コマンドを利用する方法
- GitHub 上で作成する方法

## リポジトリの clone

- Web ブラウザで GitHub の URL にアクセス
- ページに表示されている「SSH clone URL」を探す

## 端末上で git clone コマンドを実行する

```
git clone <SSH clone URL>
```

# 基本的な git コマンド

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行う  
GitHub

TODO 補  
足資料

## 新しくブランチを作成してチェックアウトする

```
git checkout -b some_new_feature
```

## ブランチを GitHub に push する

```
git add .  
git commit -m '（作業内容）'  
git push -u origin some_new_feature
```

## master を最新版にする

```
git checkout master  
git pull
```



## 27 GitHub とは

## 28 ペアで行う GitHub

## 29 グループで行う GitHub

## 30 TODO 補足資料

# ペアで GitHub を使ってみよう

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

- 1 隣同士でペアを組む
- 2 レポジトリを作成する（どちらか一方）
  - `bundle gem` でひな形を作る（初心者は `Gem` でなくても良い）
- 3 レポジトリの Collaborators に登録する
- 4 レポジトリに対して、次のことを行う
  - Pull requests を利用してみる
  - Issue を利用してみる
  - Wiki を利用してみる

# 課題 1

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行う  
GitHub

TODO 補  
足資料

- 1 Pull request & merge の作業を各自 5 回以上行う
  - ディスカッションやコードレビューもやってみる
- 2 Issue を 5 個以上登録する
  - Pull request による Issue の close など試す
- 3 Wiki でページを作成する
  - ページを 5 つ程度作成して、リンクも貼る
- 4 以上が終わったペアはグループでの演習に進む
  - 講師に申告すること

## 27 GitHub とは

## 28 ペアで行う GitHub

## 29 グループで行う GitHub

## 30 TODO 補足資料

# グループで GitHub

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

- 1 ペアを 2 つ組み合わせて 4 人グループを作成する
  - 課題 1 が終わったペアから順番にグループ編成
- 2 作りたい Gem について相談する
  - テーマはなんでも良い
  - Web API を利用したコマンドラインツールなど
  - ある程度の役割分担も決めておく
- 3 レポジトリを作成する（代表者 1 名）
  - Gem を作成する
- 4 今まで学んだ知識を活用して Gem を開発する

## 課題 (2)

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

- 1 グループメンバー全員である程度のコーディングを行う
- 2 GitHub Flow を実践する
- 3 Travis CI によるテストの自動化
- 4 RubyGems.org へのディプロイ
- 5 その他、GitHub の各種機能の活用

## 27 GitHub とは

## 28 ペアで行う GitHub

## 29 グループで行う GitHub

## 30 **TODO** 補足資料

# Travis CI ステータスイメージ

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行う  
GitHub

TODO 補  
足資料

## Travis CI のテスト結果を分かりやすく共有したい

- Travis CI のステータスイメージを README.md に埋め込むことができる
  - Embedding Status Images - Travis CI
  - Travis CI と Coveralls と Code Climate を使って GitHub リポジトリにバッジを付ける - アインシュタインの電話番号



# GitHub でのコンフリクトの解消方法

フレームワーク開発特論  
(2015)

中鉢欣秀

GitHub とは

ペアで行う  
GitHub

グループで行  
う GitHub

TODO 補  
足資料

## 前提

- `new_feature` ブランチで作業中であり、最新の更新は `commit` 済

## 操作（一例）

```
git fetch origin          # origin の最新版を取得
git checkout master       # master をチェックアウト
git pull origin master    # 手元の master を最新版にする
git checkout new_feature  # 作業中のブランチに戻る
git merge master          # この後、コンフリクトを修正する
git push origin new_feature # 作業中のブランチを再度、push
```

# フレームワーク開発特論 (2015)

中鉢欣秀

産業技術大学院大学 (AIIT)

第9章まとめ

## 31 この授業で取り上げたこと

## 32 課題の提出方法

# Ruby のエコシステム (1)

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## rbenv

- Ruby そのもののバージョン管理
- 言語のインストールも自動化

## rake

- 開発で必要なタスクの自動化
- Rakefile に設定を書く

# Ruby のエコシステム (2)

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## Gem

- Ruby のコンポーネント開発
- RubyGems による公開

## bundler

- Gem のダウンロードやロードパスの設定
- Gem を開発するための機能もある

# Ruby のエコシステム (3)

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## MiniTest

- 単体テストのフレームワーク
- RSpec 等他のフレームワークも存在

## Guard

- テストの自動化
- プラグインにより、他にも多くのタスクが自動化できる

# 継続的統合

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## Travis CI

- 様々な開発環境で利用できる CI 環境
- GitHub と連携する
- Ruby の場合, Rake のデフォルトのタスク (rake test) が実行される

# GitHub

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## GitHub

- GitHub の各種機能
- ペア/グループによる開発演習



## 31 この授業で取り上げたこと

## 32 課題の提出方法

# 課題提出について

フレームワーク開発特論  
(2015)

中鉢欣秀

この授業で取り上げたこと

課題の提出方法

## 提出内容

- 1 GitHub の URL (RubyGems.org の URL は不要)
  - Gem の演習で作成したリポジトリ
  - ペア演習で作成したリポジトリ
  - グループ演習で作成したリポジトリ
- 2 レポート
  - タイトル「この授業で学んだこと」(800 字程度)
- 3 自己評価
  - 5 段階評価でいくつくらいか (2 以上合格)

## 提出先と期限

- 別途掲示板にて連絡