



Revel

A high-productivity web framework for the Go language.

Revel Web Framework 入門

堀田直孝 著

2014-05-03 版 Hungry Foolish Gray Brain. 発行

はじめに (仮)

ここに何を書こうかな ~????? 最終的には、この部分は削除になるかな ~?

ご協力をお願い:

私は、日本語が得意ではありません。誤字脱字については、是非ご連絡頂けると幸いです。又用語の定義違いその他、お気づきの点は是非ご連絡頂けると幸いです。

この本のために使ったソースコードは、Github で公開しています。pull request で修正を頂けますと、ソースの修正から再公開までの作業がスムーズに進むと思われます。是非、Github での pull request 申請による修正も活用していただけると幸いです。

Github Repository: <https://github.com/jhotta/revel-book>

謝辞:

この本のコンテンツは、Ruby on Rails チュートリアル (<http://railstutorial.jp/>) にインスパイアされて作成しました。このような素晴らしいチュートリアルを公開していただくことに感謝します。更にこの本の査読を手伝ってくれたコーワーキングスペース [mitacafe](#) のメンバーの皆さんに感謝します。

最後に、無職の私に本を書く動機とチャンスを与えてくれた妻と娘に感謝します。

筆者について



図: @jhotta

略歴

略歴は、肯定的に書こうと思っているのですが何を書いたらいいのか思い当たりません。「自分の良いところをきちんと認知しないと…」と常々思っているのですが、妄想ばかりのような気がします。

- DevOps に関して真剣の考えたことのある人材
- プログラマーの地位を向上させたいと思っている人材
- 妻のことが大好きな人材

協力者リスト

以下の協力者を紹介します。

- aaaaa
- bbnbbbb

準備する環境

この本の対象とする読者にはマイクロソフト社製の OS を利用している人も多々いると思うが、私自身 Windows XP 以降マイクロソフト社製 OS を利用していない。現状それらの OS がどのような状況になっているか把握できていないのだ。従ってこの本では、インストールや実行確認作業を仮想 OS 環境上の Linux OS で進めていくことにする。

又、基本的な操作は Apple 社製の OS から実施している。万が一この本で使っているツールがマイクロソフト社製の OS 向けに存在しない場合は、操作の目的に合ったツールに切り替え、操作を完結して欲しい。

仮想 OS 環境を実現するソフト

VirtualBox: <https://www.virtualbox.org/>

VirtualBox は、x86 仮想化ソフトウェア・パッケージの一つで、米国オラクル社によって開発がすすめられています。サポートされているホスト OS は Linux、Mac OSX、Windows、Solaris です。ゲスト OS としては、FreeBSD、Linux、OpenBSD、OS/2 Warp、Windows、Mac OS X Server、Solaris など x86/x64 アーキテクチャの OS であれば基本的には起動できます。GPL ver.2 で公開されている FOSS なので、無料で使用することが出来ます。

仮想 OS 環境で利用する Linux OS

Ubuntu 13.10: server 64bit <http://www.ubuntu.com/download/server>

Ubuntu は、Debian GNU/Linux をベースとした Linux ディストリビューションの 1 つです。Ubuntu は、「誰にでも使いやすい最新かつ安定した OS」を目標にカノニカル社から支援を受けて開発されています。毎年 4 月、10 月に更新版がリリースされ、LTS(長期メンテナンス)バージョンは 2 年に一度リリースされ、12.04 が直近バージョンになります。

仮想 OS の内容を設定するためのソフト

Vagaran: <http://www.vagrantup.com/>

Vagrant は、仮想 OS 環境を設定したり、設定後の仮想マシンのイメージを作成指定くれるオープンソースのソフトウェアです。VirtualBox や VMware などの仮想 OS 環境と Puppet, Chef, Ansible などの構成管理ツールの間を取り持って、再現性の高い管理環境を提供してくれます。

又、仮想 OS の起動, 設定, SSH 通信, マシンイメージ作成などをコマンドラインから操作することができ、開発者の操作性に高い操作性を提供しています。

目次

はじめに (仮)	i
ご協力のお願い:	i
謝辞:	i
筆者について	ii
略歴	ii
協力者リスト	iii
準備する環境	iv
第 1 章 Browser に Revel サイトの画面を表示するまで	1
1.1 ソフトウェアのインストールと仮想マシンイメージの設定	1
1.1.1 VirtualBox のインストール	1
1.1.2 Vagrant による環境設定	2
1.1.3 Go 言語のインストール	5
1.1.4 Revel web framework インストール	5
第 2 章 デモアプリケーション	6
2.1 VirtualBox と ubuntu のインストール	6
2.1.1 VirtualBox の準備	7
2.1.2 ubuntu のインストール	7
2.2 Go 言語のインストール	7
2.3 Revel web framework インストール	7
2.4 実行に必要な環境変数の設定	8
第 3 章 静的ページの作成	9
3.1 本文の書き方	9
3.1.1 見出し	9
3.2 箇条書き	9
3.3 ソースコードなどのリスト	10
3.4 画像	10
第 4 章 Revel Framework で必要な Go 言語超基礎	11

目次

4.1	本文の書き方	11
4.1.1	見出し	11
4.2	箇条書き	11
4.3	ソースコードなどのリスト	12
4.4	画像	12
第5章	レイアウトを作成する	13
5.1	本文の書き方	13
5.1.1	見出し	13
5.2	箇条書き	13
5.3	ソースコードなどのリスト	14
5.4	画像	14
第6章	ユーザーのモデルを作成する	15
6.1	本文の書き方	15
6.1.1	見出し	15
6.2	箇条書き	15
6.3	ソースコードなどのリスト	16
6.4	画像	16
第7章	ユーザー登録	17
7.1	本文の書き方	17
7.1.1	見出し	17
7.2	箇条書き	17
7.3	ソースコードなどのリスト	18
7.4	画像	18
第8章	サインイン、サインアウト	19
8.1	本文の書き方	19
8.1.1	見出し	19
8.2	箇条書き	19
8.3	ソースコードなどのリスト	20
8.4	画像	20
第9章	ユーザーの更新・表示・削除	21
9.1	本文の書き方	21
9.1.1	見出し	21
9.2	箇条書き	21
9.3	ソースコードなどのリスト	22
9.4	画像	22
第10章	ユーザーのマイクロポスト	23

目次

10.1	本文の書き方	23
10.1.1	見出し	23
10.2	箇条書き	23
10.3	ソースコードなどのリスト	24
10.4	画像	24
第 11 章	ユーザーをフォローする	25
11.1	本文の書き方	25
11.1.1	見出し	25
11.2	箇条書き	25
11.3	ソースコードなどのリスト	26
11.4	画像	26
第 12 章	Test, CI, CD	27
付録 A	asdfasdfas	28
A.1	asdfasdfasfa	28
A.1.1	34345345345345	28
A.2	hijklhkkjlhlkhjkl	28
付録 B	fadfasdfasdf	29
B.1	dfasdfasdfwasdf	29

第 1 章

Browser に Revel サイトの画面を表示するまで

Go 言語を使って Revel web framework が、起動できる環境を構築していきます。

1.1 ソフトウェアのインストールと仮想マシンイメージの設定

1.1.1 VirtualBox のインストール

この本では、VirtualBox を利用してベースとなる LinuxOS を起動することにします。VirtualBox は、以下のページ URL からダウンロードできます。

今作業している PC の OS に適応した VirtualBox のパッケージを、このサイトからダウンロードします。



図 1.1 VirtualBox のダウンロードページ

先にダウンロードした VirtualBox のインストールファイルをダブルクリックし、ポップアップメニューの指示に従って VirtualBox をインストールします。

第 1 章 Browser に Revel サイトの画面を表示するまでのインストールと仮想マシンイメージの設定

--[[path = (not exist)]]--

インストール図 1

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い VirtualBox をインストールします。

--[[path = (not exist)]]--

インストール図 2

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い VirtualBox をインストールします。

--[[path = (not exist)]]--

インストール図 3

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い VirtualBox をインストールします。

1.1.2 Vagrant による環境設定

Vagrant のインストール

VirtualBox は直接操作しても特に問題はなのですが、仮想 OS 環境を効率よく操作すると共に基本環境の構築の再現性を上げるために、今回は Vagrant を使用することにします。

Vagrant は、以下の[ページ URL](#) からダウンロードできます。

今作業している PC の OS に適応した VirtualBox のパッケージを、このサイトからダウンロードします。

--[[path = (not exist)]]--

Vagrant のダウンロードページ

先にダウンロードした VirtualBox のインストールファイルをダブルクリックし、ポップアップメニューの指示に従って VirtualBox をインストールします。

--[[path = (not exist)]]--

インストール図 1

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い Vagrant をインストールします。

第 1 章 Browser に Revel サイトの画面を表示するまでのインストールと仮想マシンイメージの設定

```
--[[path = (not exist)]]--
```

インストール図 2

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い Vagrant をインストールします。

```
--[[path = (not exist)]]--
```

インストール図 3

ダウンロードしたファイルをダブルクリックし、ポップアップメニューの指示に従い Vagrant をインストールします。

```
$ vagrant --version
```

以下の内容が表示されれば、Vagrant のインストールは終了です。

```
Vagrant 1.4.3
```

仮想マシンイメージの準備

<http://www.vagrantbox.es/>

```
--[[path = (not exist)]]--
```

インストール図 3

今回は、"Official Ubuntu 13.10 daily Cloud Image amd64 (Development release, No Guest Additions)"と書かれている仮想マシンイメージを Vagrant の管理下にインポートして使うことにします。

コマンドの基本は以下のフォーマットになります。

```
$ vagrant box add {title} {url}
```

今回は、ubuntu コミュニティーが提供している 先の仮想イメージの url を指定し、ubuntu を title に指定し、Vagrant 管理下にダウンロードしてきます。

第1章 Browser に Revel サイトの画面を表示するまでのインストールと仮想マシンイメージの設定

```
$ vagrant box add ubuntu \  
http://cloud-images.ubuntu.com/vagrant/saucy/current/  
saucy-server-cloudimg-amd64-vagrant-disk1.box
```

上記のコマンドを実行すると、下記の内容がターミナルウィンドーに表示され、登録が終了します。

```
WARN: Unresolved specs during Gem::Specification.reset:  
      ffi (>= 1.0.11, ~> 1.0)  
WARN: Clearing out unresolved specs.  
Please report a bug if this causes problems.  
Downloading box from URL: http://cloud-images.ubuntu.com/vagrant/saucy/current/saucy-server-cloudimg-amd64-vagrant-disk1.box  
Extracting box...te: 1131k/s, Estimated time remaining: --:--:--  
Successfully added box 'ubuntu' with provider 'virtualbox'!
```

ここに解説を書く！

```
$ vagrant box revmove {title}
```

ここに解説を書く！

```
$ mkdir ~/vagrant-env  
$ cd ~/vagrant-env
```

ここに解説を書く！

```
$ vagrant init {title}
```

ここに解説を書く！

```
$ vagrant init vagrant-env
```

ここに解説を書く！

第 1 章 Browser に Revel サイトの画面を表示するまでのインストールと仮想マシンイメージの設定

```
$ vagrant up
```

ここに解説を書く！

こんな内容が表示される。

ここに解説を書く！

```
$ vagrant ssh
```

ここに解説を書く！

ssh の接続結果を表示

ここに解説を書く！

Vagrantfile の準備

1.1.3 Go 言語のインストール

VirtualBox 上に駆動している ubuntu の仮想マシンへ ssh を使ってアクセスします。

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

1.1.4 Revel web framework インストール

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

第 2 章

デモアプリケーション

Go 言語と Revel web framework が、実行できる環境を構築していきます。



図: 想い

2.1 VirtualBox と ubuntu のインストール

読者の皆さんの作業内容を統一するために今回は、virtualbox に Linux OS をインストールし、その上で作業を進めることにします。

2.1.1 VirtualBox の準備

VirtualBox は、x86 仮想化ソフトウェア・パッケージの一つで、米国オラクル社によって開発がすすめられています。サポートされているホスト OS は Linux、Mac OSX、Windows、Solaris です。ゲスト OS としては、FreeBSD、Linux、OpenBSD、OS/2 Warp、Windows、Mac OS X Server、Solaris など x86/x64 アーキテクチャの OS であれば基本的には起動できます。GPL ver.2 で公開されている FOSS なので、無料で使用することが出来ます。

この本では、このソフトを利用し学習ベースとなる LinuxOS を起動することにします。VirtualBox は、以下の [URL のページ](https://www.virtualbox.org/wiki/Downloads) からダウンロードできます。目的のコンピューターの OS に適した VirtualBox のパッケージをダウンロードしインストールしていきましょう。

VirtualBox のダウンロードページ:

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

2.1.2 ubuntu のインストール

今回は、VirtualBox 上の OS に ssh で接続し作業することになります。従って、ubuntu 13.10 64bit server 版をインストールすることになります。

インストールの詳細に関しては、[こちら](#) してください。

ubuntu 13.10 server 64bit 版 ISO イメージのダウンロードページ:

<http://www.ubuntu.com/download/server>

2.2 Go 言語のインストール

VirtualBox 上に駆動している ubuntu の仮想マシンへ ssh を使ってアクセスします。

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

2.3 Revel web framework インストール

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

2.4 実行に必要な環境変数の設定

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 目
- 2 目
- 3 目

第3章

静的ページの作成

書き方のサンプルです。上書きするなりして消して下さい。

3.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2行以上以上空いていても1行空いているのと同様に処理します。

3.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

3.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1つ目
- 2つ目
- 3つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1つ目
2. 2つ目

3. 3 つ目

3.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 3.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

3.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 4 章

Revel Framework で必要な Go 言語 超基礎

書き方のサンプルです。上書きするなりして消して下さい。

4.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

4.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

4.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目
3. 3 つ目

4.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 4.1: リストのサンプル

```
int main(int argc, char **argv) {
    puts("OK");
    return 0;
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main
  puts "ok"
end
```

4.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/revel/blob/master/doc/format.rdoc> を御覧ください。

第 5 章

レイアウトを作成する

書き方のサンプルです。上書きするなりして消して下さい。

5.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

5.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

5.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 つ目

5.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 5.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

5.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 6 章

ユーザーのモデルを作成する

書き方のサンプルです。上書きするなりして消して下さい。

6.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

6.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

6.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 つ目

6.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 6.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

6.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 7 章

ユーザー登録

書き方のサンプルです。上書きするなりして消して下さい。

7.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

7.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

7.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 つ目

7.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 7.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

7.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 8 章

サインイン、サインアウト

書き方のサンプルです。上書きするなりして消して下さい。

8.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

8.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

8.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 回目

8.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 8.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

8.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第9章

ユーザーの更新・表示・削除

書き方のサンプルです。上書きするなりして消して下さい。

9.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2行以上以上空いていても1行空いているのと同様に処理します。

9.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

9.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1つ目
- 2つ目
- 3つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1つ目
2. 2つ目

3. 3 つ目

9.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 9.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

9.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 10 章

ユーザーのマイクロポスト

書き方のサンプルです。上書きするなりして消して下さい。

10.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

10.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

10.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 つ目

10.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 10.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

10.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 11 章

ユーザーをフォローする

書き方のサンプルです。上書きするなりして消して下さい。

11.1 本文の書き方

最初の段落です。この行も同じ段落です。

次の段落です。

2 行以上以上空いていても 1 行空いているのと同様に処理します。

11.1.1 見出し

「=」「==」「===」の後に一文字空白をあけると見出しになります。

コラム: コラムについて

見出しの先頭に「[column]」と書くと、そこはコラムになります。

11.2 箇条書き

番号のない箇条書きは「*」を使います。前後に空白を入れて下さい。

- 1 つ目
- 2 つ目
- 3 つ目

番号付きの箇条書きには、「1.」「2.」などと書きます。数字の値は実際には無視され、連番が振られます。

1. 1 つ目
2. 2 つ目

3. 3 つ目

11.3 ソースコードなどのリスト

リストには「`//list`」ブロックや「`//emlist`」ブロックを使います。

リスト 11.1: リストのサンプル

```
int main(int argc, char **argv) {  
    puts("OK");  
    return 0;  
}
```

文中にリストを書くには「`//emlist`」になります。

```
def main  
  puts "ok"  
end
```

11.4 画像

画像は「`//image`」ブロックを使います。

```
--[[path = (not exist)]]--
```

画像サンプル

より詳しくは、<https://github.com/kmuto/review/blob/master/doc/format.rdoc> を御覧ください。

第 12 章

Test, CI, CD

サンプルその 2

基本はサンプルと同様です。

- 2asdfa - asdfasdfa - asdfasdfs

付録 A

asdfasdfas

A.1 asdfasdfasdfa

A.1.1 34345345345345

A.2 hkljhkklijhkhjkl

* aaaaa * bbnbbbb

付録 B

fadfasdfasdf

B.1 dfasdfasdfwasdf

Revel Web Framework 入門

2014 年 05 月 03 日 v1.0.0 版発行

著 者 堀田直孝

編集者 堀田直孝

発行所 Hungry Foolish Gray Brain.

(C) 2014 Naotaka Jay Hotta