Ruby と Sinatra ではじめる Web アプリケーション開発の教科書

ソースコード

2021 年 7 月 29 日版

森北出版株式会社

目次

第 3 章

3.1 節

/.bash_profile

```
export PATH="$HOME/.rbenv/bin:$PATH"
eval "$(rbenv init -)"
```

HelloWorld.rb

```
# This is my first Ruby program
puts "Hello, World"
```

3.12 節

args.rb

```
puts "count = #{ARGV.count}"

ARGV.each do |i|
  puts i
end
```

3.14 節

curry.rb

```
class Curry
  def initialize(a = 3, b = 300)  #  引数が省略されたら辛さ3と量300を使う
    setKarasa(a)
    setRyou(b)
  end

  def setKarasa(a)
    @karasa = a
  end

  def setRyou(a)
    @ryou = a
  end

  def getKarasa
    return (@karasa)
  end
  def getRyou
    return (@ryou)
  end
end

iguano = Curry.new(5)        #  辛さは5, 量は指定せずデフォルト値を使う
stego  = Curry.new(1, 250)  #  辛さは1で量は250

puts "iguano"
puts "  karasa = #{iguano.getKarasa}"
puts "  ryou = #{iguano.getRyou}"
puts "stego"
puts "  karasa = #{stego.getKarasa}"
puts "  ryou = #{stego.getRyou}"
```

curry.rb_A

```
class Curry
  def initialize(a = 3, b = 300)
    setKarasa(a)
    setRyou(b)
  end

  def setKarasa(a)
    @karasa = a
  end

  def setRyou(a)
    @ryou = a
  end

  def getKarasa
    return (@karasa)
  end

  def getRyou
    return (@ryou)
  end
end
```

curry.rb_B

```
require './curry_A'  #  curry_A.rb(=Curryクラスの定義)を読み込む

iguano = Curry.new(5)
stego  = Curry.new(1, 250)
```

```
puts "iguano"
puts "  karasa = #{iguano.getKarasa}"
puts "  ryou = #{iguano.getRyou}"
puts "stego"
puts "  karasa = #{stego.getKarasa}"
puts "  ryou = #{stego.getRyou}"
```

3.15 節

kodomocurry.rb

```
class Curry
  def initialize(a = 3, b = 300)
    setKarasa(a)
    setRyou(b)
  end

  def setKarasa(a)
    @karasa = a
  end

  def setRyou(a)
    @ryou = a
  end

  def getKarasa
    return (@karasa)
  end

  def getRyou
    return (@ryou)
  end
end


class KodomoCurry < Curry
  def initialize(a = 1, b = 200)  # 甘口でご飯は少なめ
    @karasa = a
    @ryou = b
  end

  def openOmake
    puts "Omake Open!"
  end
end
```

```
iguano = Curry.new(5)
stego = KodomoCurry.new  #  KodomoCurryクラスのインスタンス

iguano.setKarasa(99)

puts "iguano"
puts "  karasa = #{iguano.getKarasa}"
puts "  ryou = #{iguano.getRyou}"

#  25～34行目には書かれていないメソッドはCurryクラスからそのまま引き継ぐ
stego.setKarasa(99)

puts "stego"
puts "  karasa = #{stego.getKarasa}"
puts "  ryou = #{stego.getRyou}"
stego.openOmake  #  stegoはopenOmakeメソッドが使える
```

3.17 節

calendar.rb

```
# 西暦y年が閏年ならtrue、平年ならfalseを返す
def isLeapYear(y)
  ...
  return(...)
end

y = 2020  # 西暦年

if isLeapYear(y) == true
  puts "西暦#{y}年は閏年です。"
else
  puts "西暦#{y}年は平年です。"
end
```

calendar.rb(*getLastDay*() を追加)

```
# 西暦y年が閏年ならtrue、平年ならfalseを返す
def isLeapYear(y)
  ...
  return(...)
end



# 西暦y年m月が何日まであるかを返す
def getLastDay(y, m)
  ...
  return(...)
end


y = 2020  # 西暦年
m = 2     # 月

dd = getLastDay(y, m)

puts "西暦#{y}年#{m}月は#{dd}日まであります。"
```

calendar.rb(*zeller*() を追加)

```
# 西暦y年が閏年ならtrue、平年ならfalseを返す
def isLeapYear(y)
  ...
  return(...)
end


# 西暦y年m月が何日まであるかを返す
def getLastDay(y, m)
  ...
  return(...)
end


# y年m月d日が何曜日であるかを0～6で返す
def zeller(y, m, d)
  ...
  return(...)
end


y = 2020
m = 3

w = zeller(y, m, 1)  # y年m月1日の曜日(0=日曜日)

puts "西暦#{y}年#{m}月1日は#{w}曜日です。"
```

calendar.rb（完成）

```
y = 2020
m = 3

g = getLastDay(y, m)  # その月が何日まであるか
```

```
w = zeller(y, m, 1)   # その月の1日は何曜日か

puts "西暦#{y}年#{m}月"
puts "Sun Mon Tue Wed Thu Fri Sat"

# 1日の前の曜日を飛ばす
c = 0
while c < w
  print "    "  # 半角4文字分の空白
  c = c + 1
end

# 続きから月末日まで書き出す
e = 1
while e <= g
  print " %2d "%e  # 2桁に揃えて変数eを表示する
  e = e + 1
  c = c + 1
  if (c % 7 == 0)  # 7日ごとに改行を入れる
    print "\n"
  end
end

print "\n"  # 月末日の後にも改行しておく
```

第 4 章

4.3 節

Gemfile(デフォルト)

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

# gem "rails"
```

Gemfile(書き換え後)

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

gem "sinatra"  # ここを書き換える
gem "webrick" # こちらも追加
```

hello.rb

```
require 'sinatra'

set :environment, :production

get '/' do
  "Hello, A Whole New World."
end
```

4.4 節

./views/layout.erb

```
<html>

<head>
<title>A famous song</title>
</head>

```

```
<body>
<h1>A famous song</h1>
<%= yield %>
</body>

</html>
```

./views/index.erb

```
Twinkle, twinkle, little star,<br>
How I wonder what you are.
```

hello.rb

```
require ’sinatra’

set :environment, :production

get ’/’ do
  erb :index
end
```

4.5 節

./views/index.erb

```
Twinkle, twinkle, little star,<br>
How I wonder <%= @msg %>.
```

hello.rb

```
require ’sinatra’

set :environment, :production

get ’/’ do
  @msg = "what you are".upcase
  erb :index
end
```

4.7 節

webcal.rb

```
require ’sinatra’
```

```
set :environment, :production

get '/:y/:m' do
  @year = params[:y]   # URLの /:y/:m の :y の部分を取得する
  @month = params[:m]  # 同じく :m の部分を取得する

  erb :moncal
end
```

./views/moncal.erb

```
<%= @year %>年 <%= @month %>月
<p>
```

webcal.rb

```
require 'sinatra'

set :environment, :production

get '/:y/:m' do
  @year = params[:y].to_i
  @month = params[:m].to_i

  @t = "<table border>"
  @t = @t + "<tr><th>Sun</th><th>Mon</th><th>Tue</th><th>Wed</th>"
  @t = @t + "<th>Thu</th><th>Fri</th><th>Sat</th></tr>"

  l = getLastDay(@year, @month)
  h = zeller(@year, @month)

  d = 1
  6.times do |p|
    @t = @t + "<tr>"
    7.times do |q|
      if p == 0 && q < h
        @t = @t + "<td></td>"  # 1行目の日曜日から1日までは空欄
      else
        if d <= l
          @t = @t + "<td>#{d}</td>"
          d += 1
        else
          @t = @t + "<td></td>"  # 月末日以降、土曜日までは空欄
        end
      end
    end
    @t = @t + "</tr>"
    if d > 1
      break
    end
  end

  @t = @t + "</table>"

  erb :moncal
end


def isLeapYear(y)
   :
end


def getLastDay(y, m)
   :
end


def zeller(y, m, d)
   :
end
```

./views/moncal.erb

```
<%= @year %>年 <%= @month %>月
<p>
<%= @t %>
```

第 5 章

5.3 節

dbinit.sq3

```
create table bbsdata (
  id integer primary key,
  name varchar(30),
  entry varchar(150)
);

insert into bbsdata values (1, ’Diplodocus’, ’The first entry.’);
insert into bbsdata values (2, ’Allosaurus’, ’The second entry.’);
```

5.4 節

Gemfile

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

gem "activerecord"
gem "sqlite3"
```

5.5 節

database.yml

```
development:
  adapter: sqlite3
  database: bbs.db
```

printyml.rb

```
require ’yaml’

d = YAML.load_file(’database.yml’)
puts d
```

5.6 節

dbtest.rb

```
require ’active_record’

# データベースを使う設定
ActiveRecord::Base.configurations = YAML.load_file(’database.yml’)
ActiveRecord::Base.establish_connection :development

# bbsdataテーブルをBBSdataクラスとして扱えるようにする
class BBSdata < ActiveRecord::Base
  self.table_name = ’bbsdata’
end

# IDが1のものを探してaに入れる
a = BBSdata.find(1)

# 見つけたレコードの各フィールドを表示
puts "#{a.id}, #{a.name}, #{a.entry}"
```

5.7 節

Gemfile

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

gem "activerecord"
gem "sqlite3"
gem "sinatra"
gem "webrick"
```

dbtest.rb

```
require ’sinatra’
require ’active_record’

set :environment, :production

ActiveRecord::Base.configurations = YAML.load_file(’database.yml’)
ActiveRecord::Base.establish_connection :development

```

```
class BBSdata < ActiveRecord::Base
  self.table_name = ’bbsdata’
end


get ’/’ do
  t = BBSdata.all

  @h = ""
  t.each do |a|
    @h = @h + "<tr>"
    @h = @h + "<td>#{a.id}</td>"
    @h = @h + "<td>#{a.name}</td>"
    @h = @h + "<td>#{a.entry}</td>"
    @h = @h + "</tr>\n"
  end

  erb :index
end
```

./views/layout.erb

```
<html>
<head>
<title>BBS</title>
</head>

<body>
<%= yield %>
</body>
</html>
```

./views/index.erb

```
<table border>

<tr>
<th>ID</th>
<th>Name</th>
<th>Entry</th>
</tr>

<%= @h %>

</table>
```

5.9 節

./views/index.erb

```
<table border>

<tr>
<th>ID</th>
<th>Name</th>
<th>Entry</th>
</tr>

<%= @h %>

<form method="post" action="/new">
<tr>
<td><input type="text" name="id"></td>
<td><input type="text" name="name"></td>
<td><input type="text" name="entry"></td>
<td><input type="submit" value="Go"></td>
</tr>
</form>

</table>
```

dbtest.rb

```
require ’sinatra’
require ’active_record’

set :environment, :production

ActiveRecord::Base.configurations = YAML.load_file(’database.yml’)
ActiveRecord::Base.establish_connection :development

class BBSdata < ActiveRecord::Base
  self.table_name = ’bbsdata’
end


get ’/’ do
  t = BBSdata.all

  @h = ""
  t.each do |a|
    @h = @h + "<tr>"
    @h = @h + "<td>#{a.id}</td>"
    @h = @h + "<td>#{a.name}</td>"
    @h = @h + "<td>#{a.entry}</td>"
    @h = @h + "</tr>\n"
  end

  erb :index
end


post '/new' do
  s = BBSdata.new
  s.id = params[:id]
  s.name = params[:name]
  s.entry = params[:entry]
  s.save
  redirect '/'
end
```

5.10 節

dbtest.rb

```
require 'sinatra'
require 'active_record'

set :environment, :production

ActiveRecord::Base.configurations = YAML.load_file('database.yml')
ActiveRecord::Base.establish_connection :development

class BBSdata < ActiveRecord::Base
  self.table_name = 'bbsdata'
end


get '/' do
  t = BBSdata.all

  @h = ""
  t.each do |a|
    @h = @h + "<tr>"
    @h = @h + "<td>#{a.id}</td>"
    @h = @h + "<td>#{a.name}</td>"
    @h = @h + "<td>#{a.entry}</td>"

    @h = @h + "<form method=\"post\" action=\"/del\">"
    @h = @h + "<td><input type=\"submit\" value=\"Delete\"></td>"
    @h = @h + "<input type=\"hidden\" name=\"id\" value=\"#{a.id}\">"
    @h = @h + "<input type=\"hidden\" name=\"_method\" value=\"delete\">"
    @h = @h + "</form>"

    @h = @h + "</tr>\n"
  end

  erb :index
end


post ’/new’ do
  s = BBSdata.new
  s.id = params[:id]
  s.name = params[:name]
  s.entry = params[:entry]
  s.save
  redirect ’/’
end


delete ’/del’ do
  s = BBSdata.find(params[:id])
  s.destroy
  redirect ’/’
end
```

第 6 章

6.1 節

hashtest.rb

```
require 'digest/sha1'  # ハッシュ値の計算に必要なgem を読み込む
require 'digest/sha2'
require 'digest/md5'

s = "Twinkle, twinkle, little star, How I wonder what you are."
puts s

puts Digest::SHA256.digest(s)      # ハッシュ値そのものを出力
puts Digest::SHA256.hexdigest(s)  # ハッシュ値の16 進表記を出力
```

6.3 節

passwd_dbinit.sq3

```
create table accounts (
  id char(20) primary key,
  salt varchar(40),
  hashed varchar(70),
  algo char(5)
);
```

genpass.rb

```
require 'digest/sha2'
require 'active_record'

ActiveRecord::Base.configurations = YAML.load_file('database.yml')
ActiveRecord::Base.establish_connection :development

class Account < ActiveRecord::Base
end

# 基本的な情報
username = "coelacanth"
rawpasswd = "ikitakaseki"
algorithm = "1"
r = Random.new
salt = Digest::SHA256.hexdigest(r.bytes(20))
hashed = Digest::SHA256.hexdigest(rawpasswd + salt)

```

```
puts "salt = #{salt}"
puts "username = #{username}"
puts "raw password = #{rawpasswd}"
puts "algorithm = #{algorithm}"
puts "hashed passwd = #{hashed}"

# データベースを更新する
s = Account.new
s.id = username
s.salt = salt
s.hashed = hashed
s.algo = algorithm
s.save

# データベースの中身をすべて出力する
@s = Account.all
@s.each do |a|
  puts ">> " + a.id + "\t" + a.salt + "\t" + a.hashed + "\t" + a.algo
end
```

database.yml

```
development:
  adapter: sqlite3
  database: passwd.db
```

6.4 節

checkpass1.rb

```
require 'digest/sha2'
require 'active_record'

ActiveRecord::Base.configurations = YAML.load_file('database.yml')
ActiveRecord::Base.establish_connection :development

class Account < ActiveRecord::Base
end

# ユーザーがキーボードから入力した文字列
trial_username = "coelacanth"
trial_passwd = "ikitakaseki"

# データベースに保存された内容を取り出す
a = Account.find(trial_username)
db_username = a.id
db_salt = a.salt
```

```
db_hashed = a.hashed
db_algo = a.algo

# ハッシュ値を計算する
if db_algo == "1"
  trial_hashed = Digest::SHA256.hexdigest(trial_passwd + db_salt)
else
  puts "Unknown algorithm is used for user #{trial_username}."
  exit(-2)
end

# 確認のため，変数をすべて表示
puts "--- DB ---"
puts "username = #{db_username}"
puts "salt = #{db_salt}"
puts "algorithm = #{db_algo}"
puts "hashed passwd = #{db_hashed}"
puts ""
puts "--- TRIAL ---"
puts "username = #{trial_username}"
puts "passwd = #{trial_passwd}"
puts "hashed passwd = #{trial_hashed}"
puts ""

# ログインの成否を確認
if db_hashed == trial_hashed
  puts "Login Success"
else
  puts "Login Failure"
end
```

checkpass2.rb

```
require ’digest/sha2’
require ’active_record’

ActiveRecord::Base.configurations = YAML.load_file(’database.yml’)
ActiveRecord::Base.establish_connection :development

class Account < ActiveRecord::Base
end

# ユーザーがキーボードから入力した文字列
trial_username = "latimeria"
trial_passwd = "ChalumnaRiver"

# データベースに保存された内容を取り出す
begin
```

```
  a = Account.find(trial_username)
  db_username = a.id
  db_salt = a.salt
  db_hashed = a.hashed
  db_algo = a.algo
rescue => e
  puts "User #{trial_username} is not found."
#  puts e.message
  exit(-1)
end

# ハッシュ値を計算する
if db_algo == "1"
  trial_hashed = Digest::SHA256.hexdigest(trial_passwd + db_salt)
else
  puts "Unknown algorithm is used for user #{trial_username}."
  exit(-2)
end

# 確認のため，変数をすべて表示
puts "--- DB ---"
puts "username = #{db_username}"
puts "salt = #{db_salt}"
puts "algorithm = #{db_algo}"
puts "hashed passwd = #{db_hashed}"
puts ""
puts "--- TRIAL ---"
puts "username = #{trial_username}"
puts "passwd = #{trial_passwd}"
puts "hashed passwd = #{trial_hashed}"
puts ""

# ログインの成否を判定
if db_hashed == trial_hashed
  puts "Login Success"
else
  puts "Login Failure"
end
```

第 7 章

7.3 節

sessiontest.rb

```
require ’sinatra’

set :sessions,
  secret: ’xxx’

set :environment, :production

get ’/page1’ do
  session[:message] = ’ABC’
  redirect ’/page2’
end

get ’/page2’ do
  session[:message]
end
```

7.4 節

decodecookie.rb

```
require ’cgi’
require ’base64’
require ’openssl’
require ’sinatra’

s = "..."

sb64, digest = CGI.unescape(s).split("--")
puts Marshal.load(Base64.decode64(sb64))
puts OpenSSL::HMAC.hexdigest(OpenSSL::Digest::SHA1.new, "xxx", sb64)
puts digest
```

第 8 章

8.1 節

jmail.rb

```
require ’sinatra’

set :environment, :production

set :sessions,
  expire_after: 7200,
  secret: ’abcdefghij0123456789’


get ’/’ do
  redirect ’/login’
end


get ’/login’ do
  erb :login
end


post ’/auth’ do
  username = params[:uname]
  pass = params[:pass]

  if ((username == "foo") && (pass == "bar"))
    session[:login_flag] = true
    session[:testdata] = "Brontosaurus"
    redirect ’/contentspage’
  else
    session[:login_flag] = false
    redirect ’/failure’
  end
end


get ’/contentspage’ do
  if (session[:login_flag] == true)
    @a = session[:testdata]
    erb :contents
  else
    erb :badrequest
  end
end
```

```
get '/logout' do
  session.clear
  erb :logout
end


get '/failure' do
  erb :failure
end
```

./views/layout.erb

```
<html>
<head>
<title>Jmail</title>
</head>

<body>
<center>
<h2>Jmail</h2>
</center>

<%= yield %>

</body>
</html>
```

./views/contents.erb

```
This is main contents page.<br>
Test data = <%= @a %><br>

<a href="/logout">logout</a>
```

./views/login.erb

```
Enter your username and password.<br>

<form action="/auth" method="post">
Username: <input type="text" name="uname" size="40" maxlength="20"><br>
Password: <input type="password" name="pass" size="40" maxlength="30"><br>
<br>
<input type="submit" value="Login">
<input type="reset" value="Reset">
```

```
</form>
```

./views/logout.erb

```
User has logged out.<br>

<a href="/login">Back to login screen.</a>
```

./views/loginfailure.erb

```
Login failed.<br>

<a href="/login">Back to login screen.</a>
```

./views/badrequest.erb

```
Bad request. Please login this service first.<br>

<a href="/login">Back to login screen.</a>
```

8.2 節

encodecookie.rb

```
require ’cgi’
require ’base64’
require ’openssl’
require ’sinatra’

s = "..."

sb64, digest = URI.decode(s).split("--")
t = Marshal.load(Base64.decode64(sb64))
t["testdata"] = "Apatosaurus"

a = Base64.encode64(Marshal.dump(t))
b = OpenSSL::HMAC.hexdigest(OpenSSL::Digest::SHA1.new, "abcdefghij0123456789", a)
c = URI.escape(a + "--" + b)
puts c
```

8.3 節

simplebbs.rb

```
require 'sinatra'

set :environment, :production


get '/' do
  redirect '/login'
end


get '/login' do
  erb :login
end
```

./views/layout.erb

```
<html>
<head>
<title>Simple BBS</title>
</head>

<body>
<center><h2>Simple BBS</h2></center>
<p>
<%= yield %>
</body>
</html>
```

./views/login.erb

```
Enter your username and password.<br>

<form action="/auth" method="post">
Username: <input type="text" name="uname" size="30" maxlength="20"><br>
Password: <input type="password" name="pass" size="30" maxlength="20"><br>
<br>
<input type="submit" value="Login">
<input type="reset" value="Reset">
</form>
```

dbinit.sq3

```
create table bbsdata (
  id integer primary key,
  userid varchar(20),
  entry varchar(150),
  writedate integer
);


create table account (
  id varchar(20) primary key,
  hashed varchar(70),
  salt varchar(10)
);

insert into account values ('diplo', '7
fba80a3642579984776939f13f769254fe1db36e9ca41d7b598e2c1d93ec52a', 'PQ');
insert into account values ('allo', '
d12354b521bdc4bcd0de3959a378176112ef3e4d5343ba887c276d13d6d76e61', 'RS');
```

database.yml

```
development:
  adapter: sqlite3
  database: bbs.db
```

./views/logout.erb

```
User has logged out.<br>

<a href="/login">Back to login screen.</a>
```

./views/loginfailure.erb

```
Login failed.<br>

<a href="/login">Back to login screen.</a>
```

./views/contents.erb

```
Hello, <%= @u %> (<a href="/logout">logout</a>)
<p>

<table>
<%= @t %>
</table>
```

./views/badrequest.erb

```
Bad request. Please login this service first.<br>

<a href="/login">Back to login screen.</a>
```

simplebbs.rb（完成版）

```
require 'sinatra'
require 'active_record'
require 'digest/sha2'

set :environment, :production

set :sessions,
  expire_after: 7200,
  secret: 'abcdefghij0123456789'

ActiveRecord::Base.configurations = YAML.load_file('database.yml')
ActiveRecord::Base.establish_connection :development


class BBSdata < ActiveRecord::Base
  self.table_name = 'bbsdata'
end


class Account < ActiveRecord::Base
  self.table_name = 'account'
end


get '/' do
  redirect '/login'
end


get '/login' do
  erb :login
end


get '/logout' do
  session.clear
  erb :logout
end


post '/auth' do
```

```
  user = params[:uname]
  pass = params[:pass]
  r = checkLogin(user, pass)
  if r == 1
    session[:username] = user
    redirect ’/contents’
  end

  redirect ’/loginfailure’
end


get ’/loginfailure’ do
  session.clear
  erb :loginfailure
end


get ’/badrequest’ do
  session.clear
  erb :badrequest
end


get ’/contents’ do
  @u = session[:username]
  if @u == nil
    redirect ’/badrequest’
  end

  @t = ""

  a = BBSdata.all
  if a.count == 0
    @t = "<tr><td>No entries in this BBS.</td></tr>"
  else
    a.each do |b|
      @t = @t + "<tr>"
      @t = @t + "<td>#{b.id}</td>"
      @t = @t + "<td>#{b.userid}</td>"
      @t = @t + "<td>#{Time.at(b.writedate)}</td>"
      if b.userid == @u
        @t = @t + "<td><form action=\"/delete\" method=\"post\">"
        @t = @t + "<input type=\"text\" value=\"#{b.id}\" name=\"id\" hidden>"
        @t = @t + "<input type=\"hidden\" name=\"_method\" value=\"delete\">"
        @t = @t + "<input type=\"submit\" value=\"Delete\"></form></td>"
      else
        @t = @t + "<td></td>"
      end
      @t = @t + "</tr>"
      @t = @t + "<tr><td colspan=\"4\">#{b.entry}</td></tr>\n"
    end
  end

  erb :contents
end


post ’/new’ do
  maxid = 0
  a = BBSdata.all
  a.each do |b|
    if b.id > maxid
      maxid = b.id
    end
  end

  s = BBSdata.new
  s.id = maxid + 1
  s.userid = session[:username]
  s.entry = params[:entry]
  s.writedate = Time.now.to_i
  s.save
  redirect ’/contents’
end


delete ’/delete’ do
  s = BBSdata.find(params[:id])
  s.destroy
  redirect ’/contents’
end


def checkLogin(trial_username, trial_password)
  r = 0  # 0=ログイン失敗を表す

  begin
    a = Account.find(trial_username)
    db_username = a.id
    db_salt = a.salt
    db_hashed = a.hashed
    trial_hashed = Digest::SHA256.hexdigest(trial_password + db_salt)

    if trial_hashed == db_hashed
      r = 1  # ログイン成功
    end
  rescue => e
```

```
    r = 2  # 未知のユーザー
  end

  return(r)
end
```

./views/contents.erb（完成版）

```
Hello, <%= @u %> (<a href="/logout">logout</a>)
<p>

<table>
<%= @t %>
</table>

<hr>

<form method="post" action="/new">
<input type="text" name="entry" size=50 maxlength=100><br>
<input type="submit" value="Go">
</form>
```

sanitizing.html

```
<html>
<body>
&lt;h1&gt;An example of XSS&lt;/h1&gt;
</body>
</html>
```

8.4 節

up.rb

```
require ’sinatra’

set :environment, :production


get ’/’ do
  images_name = Dir.glob("public/files/*")
  @images_path = []
  images_name.each do |a|
    @images_path << a.gsub("public/files/", "")
  end
  erb :index
```

```
end


post '/upload' do
  s = params[:file]
  if s != nil
    save_path = "./public/files/#{params[:file][:filename]}"
    File.open(save_path, 'wb') do |f|
      g = params[:file][:tempfile]
      f.write g.read
    end
  else
    puts "Upload failed"    # アップロード失敗
  end
  redirect '/'
end
```

./views/index.erb

```
<% @images_path.each do |b| %>
<a href="/files/<%= b %>"><%= b %></a><br>
<% end %>
<p>
<form action="/upload" method="post" enctype="multipart/form-data">
<input type="file" name="file">
<input type="submit" name="submit">
</form>
```

8.6 節

*dbinit.sq*3

```
create table bbsdata (
  id integer primary key,
  userid varchar(20),
  entry varchar(150),
  imgfile varchar(260),  /* アップロードした画像ファイル名 */
  writedate integer
);


create table account (
  id varchar(20) primary key,
  hashed varchar(70),
  salt varchar(10),
  iconfile varchar(260)  /* アイコン画像ファイル名 */
);
```

```
insert into account values ('diplo', '7
fba80a3642579984776939f13f769254fe1db36e9ca41d7b598e2c1d93ec52a', 'PQ', 'penguin.
jpg');
insert into account values ('allo', '
d12354b521bdc4bcd0de3959a378176112ef3e4d5343ba887c276d13d6d76e61', 'RS', 'bench.jpg
');
```