

Started on Thursday, 24 February 2022, 1:06 PM

State Finished

Completed on Thursday, 24 February 2022, 2:06 PM

Time taken 59 mins 49 secs

Grade 44.50 out of 100.00

Feedback 期末試験、お疲れ様でした。

Question **1**

Incorrect

Mark 0.00 out of 2.00

次の会話からBの言う「大丈夫」の意味に該当するのはどれですか

会話

A：「そのタスクの納期は今日中だけど、手伝いましょうか？」

B：「いえ、大丈夫です」

- ☐ 1. OK
- ☐ 2. 手伝いは不要です
- ☒ 3. 間に合います
- ☐ 4. ありがとうございます

✓

Question **2**

Incorrect

Mark 0.00 out of 2.00

次の文章で数値化または数式化できないものはどれですか。

- ☒ 1. データ量制限は100GB未満
- ☐ 2. 100GBを超えたら自動的にアーカイブする
- ☒ 3. 十分なデータ量制限
- ☒ 4. データ量制限は100GB未満

✗

✓

✗

Question **3**

Incorrect

Mark 0.00 out of 2.00

次の表現の中で数値化または数式化できない表現はどれですか。

- ☐ 1. 明日以降
- ☐ 2. ちょうどいいところ
- ☐ 3. ここからそこまで
- ☒ 4. Windows10 Professional Editionの最大メモリ容量

✖

Question **4**

Incorrect

Mark 0.00 out of 2.00

次のストーリーのなかで3つのPについて、正しく示しているのはどれですか。

ストーリー

私にはよく勉強を教えてくれるおじさんがいます。おじさんは教え方がうまく、私はおじさんのお陰でこの大学に入ることができました。

おじさんの教え方の秘訣のひとつに

「覚えられないことはとにかく何度も書いてみなさい」ということでした。

私は字が下手なので書くのが嫌でした。それでも何度も書いてみると、確かにいつの間にか覚えていました。

今日は、そんなおじさんの教え方について、もう少し詳しくお話ししたいと思います。

- ☒ 1. P: 話の前提は「おじさんの教え方」
- ☐ 2. P: 話の展開は「私は字が下手」
- ☒ 3. P: 登場人物は「おじさん」
- ☐ 4. P: 話の展開は「おじさんのお陰でこの大学に入れたこと」

✖

✔

Question **5**

Incorrect

Mark 0.00 out of 2.00

ジェスチャーを入れることのメリットについての説明です。

不適切なものはどれですか。

- ☐ 1. ジェスチャーを入れることで、聞き手の注意を分散することができます
- ☒ 2. ジェスチャーを入れることで、視覚情報にも訴えることができます
- ☐ 3. ジェスチャーを入れることで、話す言葉を省略することができます

✖

Question **6**

Incorrect

Mark 0.00 out of 2.00

「はい、いいえ」で答えられる質問を作るうえで、不適切な記述はどれですか。

- ☐ 1. 真の原因は絶対に一つでなければならない
- ☐ 2. 最初の質問は現象（悪いこと）を引き起こす「問題」を分解する質問がよい
- ☒ 3. 解決するアイデアが出そうにない原因は無視してもよい ✓
- ☒ 4. 必ず、最初の問題から順番にたどって行って質問を作らなければならない ✓

Question **7**

Incorrect

Mark 0.00 out of 2.00

自責はどれですか。

- ☒ 1. バスが遅れた ✗
- ☐ 2. バスを乗り過ごした
- ☐ 3. バスに乗り遅れた
- ☐ 4. バスが故障した

Question **8**

Correct

Mark 2.00 out of 2.00

次の文章の空欄に入る助詞を選び、それぞれの空欄にドラッグ&ドロップしなさい。

この状態では納期に間に合わない。

納期に間に合わせるには、

他のチーム ✓ 二人のプログラマー ✓

応援に来てもらわなければならない。

から のに に まで を が

Question **9**

Incorrect

Mark 0.00 out of 2.00

間違った表現はどれですか。

- ☒ 1. 経験と知識がもっとできます
- ☐ 2. インターネットはますます発展です
- ☒ 3. 仕事を役割します
- ☐ 4. 利便性が高い



Question **10**

Incorrect

Mark 0.00 out of 2.00

5W1Hが不明なものはどれですか

(5W1Hのうち、共通認識としてお互いに当然わかる項目は省略されています)

- ☐ 1. 今日の12時にグループで、いつもの店でランチミーティングをします
- ☐ 2. 明日から始まるプロジェクトのキックオフを行います
- ☒ 3. いつ、どこで、だれが、なにを、どのようにしてしますか、そしてなぜしますか？
- ☐ 4. 明日の午後3時に、大学の正門の前に集合です



Question **11**

Incorrect

Mark 0.00 out of 2.00

わかりやすいスライドを作るための説明です。

正しい記述はどれですか

- ☐ 1. 改行して単語が途中でできてしまうときは、ハイフンをつけるとよい
- ☐ 2. 1行が「す。」だけになってしまうような極端に短い行をつくらない
- ☒ 3. 単語や言葉のかたまりの途中でできると読みにくくストレスになるので、なるべく切らないほうがよい
- ☐ 4. 改行位置は、単語の途中で切れたとしても行末をそろえた方がきれいなので行末で改行する



Question 12

Incorrect

Mark 0.00 out of 2.00

次の状況を読んで、悪いこと（現象）を引き起こす「問題」はどれですか。

この問の答えはひとつです。

場面

世界中で新型コロナウイルスの感染爆発が起きています。

各国政府は海外からの入国を制限し、国内ではロックダウン政策を実施して感染拡大を防止しています。我々大学生にとっても大学内に入ることはできず、授業もすべてオンラインとなっています。

しかしオンライン授業ではネットワーク環境や停電などで授業を完全に受けられない時もあります。

だから、わからないところがたくさんあります。

- ☒ a. ネットワーク環境や停電で授業を完全に受けられない
- ☐ b. 大学内に入ることができず、すべてオンライン授業
- ☐ c. わからないところがたくさんある
- ☒ d. 新型コロナウイルスの感染爆発

✓

✗

Question 13

Incorrect

Mark 0.00 out of 2.00

問題解決できない人がとる考えや行動はどれですか。

- ☐ 1. 問題が起きても早く逃げれば大丈夫
- ☐ 2. 原因がわからなければ、考え方を変える
- ☒ 3. 自分で考えず、人からの指示を待っている
- ☒ 4. 自分から何かをやってみるということがない

✓

✓

Question 14

Correct

Mark 2.00 out of 2.00

IT用語しりとりです。（単語のみ：2語以上の複合は不可）

次の言葉の後にふさわしい答えはどれですか。

「トークンリング」 - 「グリッド」 - 「ドキュメント」 - 「

- ☐ 1. トークンリング
- ☒ 2. トラッキング
- ☐ 3. トロイの木馬

✓

Question **15**

Incorrect

Mark 0.00 out of 2.00

解決案をたくさん出しました。

そのうち2案を実行する解決策として決定したい。

決める要素として**不適切なもの**はどれですか

- ☐ 1. これは、お金がかからなくていいや
- ☐ 2. これは、友達がやってくれるから楽でいいや
- ☒ 3. これならすぐに効果が出そうだから、あとは楽だね
- ☐ 4. これは、楽に実行できそうだなあ

✖

Question **16**

Incorrect

Mark 0.00 out of 2.00

Ruby on Railsの設計方法のメリットとして、適切なものを選びなさい。

- ☐ 1. 専門性の高い業務を分けることで、作業工数が増えます。
- ☐ 2. 専門的な仕事の独立性が非常に高く、変更などの対応にも柔軟に行なえます。
- ☒ 3. 基本的なページの表示を該当するファイルに分けることで、作業工数が増えます。
- ☐ 4. 専門性の高い業務を分けることで、コードをどこに分類すれば良いのか考える時間があります。

✖

Question **17**

Correct

Mark 2.00 out of 2.00

Ruby on Railsの特徴として不適切なものを選びなさい。

- ☐ 1. Scaffolding
- ☐ 2. Migration
- ☐ 3. ActiveRecord
- ☒ 4. Yarn

✔

Question **18**

Correct

Mark 2.00 out of 2.00

ルーティングを確認するコマンドで正しいものを選びなさい。

- ☐ 1. rails routes
- ☐ 2. rake routing
- ☒ 3. rake routes
- ☐ 4. rails routing



Question **19**

Correct

Mark 2.00 out of 2.00

マイグレーションスクリプトを元にDBテーブルを作るためのコマンドとして、正しいものを選びなさい。

- ☐ 1. rake database:migration
- ☒ 2. rake db:migrate
- ☐ 3. rake db:migration
- ☐ 4. rake database:migrate



Question **20**

Correct

Mark 2.00 out of 2.00

Railsサーバーを起動するコマンドとして適切なものを一つ選びなさい。

- ☒ 1. rails s
- ☐ 2. rails c
- ☐ 3. rails a
- ☐ 4. rails g



Question **21**

Incorrect

Mark 0.00 out of 2.00

WelcomeControllerで利用される可能性のあるlayoutテンプレートとして適切なものを一つ選びなさい。

- ☐ app/views/layouts/welcome.html.erb
- ☐ app/views/welcome/welcome.html.erb
- ☒ app/views/welcome/layout.html.erb
- ☐ app/views/layouts/layout.html.erb

✗

Question **22**

Correct

Mark 2.00 out of 2.00

サインインしているユーザーのみ、アクションにアクセス可能にするために コントローラーに追加するコードとして、正しいものを選びなさい。

* Devise を使用しているものとする。

- ☐ 1. before_action :authenticate_user!
- ☐ 2. after_action authenticate_user!
- ☐ 3. after_action user_signed_in?
- ☒ 4. before_action :user_signed_in?

✗

Question **23**

Correct

Mark 2.00 out of 2.00

レビュー 1 つにつき1人のユーザを扱えるように、レビューのモデルファイルに追加するコードとして、正しいものを選びなさい。

- ☒ 1. belongs_to :user
- ☐ 2. belongs_to :user!
- ☐ 3. belongs_to :user_signed_in!
- ☐ 4. belongs_to :user_signed_in

✓

Question **24**

Correct

Mark 2.00 out of 2.00

次のテーブル構造があるとき、Scaffoldingのコマンドで正しいものを選びなさい。

テーブル名: images

カラム	日本語	データ型
user_id	ユーザID	数値型
file	画像ファイル	文字列 (255) 文字未満
created_at	作成日	日付型

- ☐ 1. rake generate scaffold images user_id:integer file:string created_at:datetime
- ☐ 2. rake g scaffold Image user_id:integer file:string created_at:datetime
- ☒ 3. rails g scaffold Image user_id:integer file:string created_at:datetime
- ☐ 4. rails generate scaffold images user_id:integer file:string created_at:datetime

Question **25**

Correct

Mark 2.00 out of 2.00

下記文章を読んで、スペースに入るものをそれぞれ選択肢の中から選んでください。

Rails を使って、ニュースにコメントできる機能を実装しています。

プロジェクトにはニュースを管理している「Newsモデル」とコメントを管理している「Commentモデル」があります。

1つのニュースに対して複数のコメントをできるようにmodelファイルを修正する場合、`has_many :comment` ファイルに `belongs_to :news` というコードを追加します。

`news.rb` `comment.rb` `has_many :comment` `has_many :news` `belongs_to :news` `belongs_to :comment`

Ruby on Railsについて以下の問いに答えなさい

授業で作成した「本のレビューアプリ」を新たに開発してみたところ、不具合（バグ）が発生しました。それぞれ不具合を確認し不具合を修正しなさい。

ソースコードを見て、一つずつの不具合に対して以下の内容を書きなさい。

- 「修正するファイル名」
- 「その行番号」
- 「修正したソースコード」
- 「不具合の理由」と「修正した理由」
(修正できるコードは1行だけです)

※ ソースコード

1. `/app/controllers/application_controller.rb`

```

1  class ApplicationController < ActionController::Base
2
3      before_action :configure_permitted_parameters, if: :devise_controller?
4
5      protected
6          def configure_permitted_parameters
7              devise_parameter_sanitizer.permit :sign_up, keys: [:name, :age, :avatar]
8              devise_parameter_sanitizer.permit :account_update, keys: [:name, :age, :avatar]
9          end
10
11  end

```

2. `/config/routes.rb`

```

1  Rails.application.routes.draw do
2      resources :book_reviews
3      get 'users/show'
4      root 'book_reviews#index'
5      devise_for :users
6      resources :books
7
8  end

```

3. `/app/views/static_pages/home.html.erb`

```

1  <h1>ホームページ</h1>
2  <%= link_to "本一覧をみる", books_path %>

```

4. /app/views/book_reviews/show.html.erb

```
1 <p id="notice"><%= notice %></p>
2
3 <p>
4   <strong>User:</strong>
5   <%= @book_review.user_id %>
6 </p>
7
8 <p>
9   <strong>Book:</strong>
10  <%= @book_review.book_id %>
11 </p>
12
13 <p>
14   <strong>Review:</strong>
15   <%= @book_review.review %>
16 </p>
17
18 <%= link_to 'Edit', edit_book_review_path(@book_review) %> |
19 <%= link_to 'Back', book_reviews_path %>
```

5. /app/models/book.rb

```
1 class Book < ApplicationRecord
2   has_one_attached :image
3
4
5   belongs_to :book_review
6 end
```

6. /app/controllers/books_controller.rb

```
1 class BooksController < ApplicationController
2   before_action :set_book, only: [:show, :edit, :update, :destroy]
3   before_action :authenticate_user!
4
5   # GET /books
6   # GET /books.json
7   def index
8     @books = Book.all
9   end
10
11  # GET /books/1
12  # GET /books/1.json
13  def show
14    @books_review = BookReview.new
15  end
16
17  # GET /books/new
18  def new
19    @book = Book.new
20  end
21
22  # GET /books/1/edit
23  def edit
24  end
25
26  # POST /books
27  # POST /books.json
28  def create
29    @book = Book.new(book_params)
30
31    respond_to do |format|
32      if @book.save
33        format.html { redirect_to @book, notice: "Book was successfully created." }
34        format.json { render :show, status: :created, location: @book }
35      else
36        format.html { render :new }
37        format.json { render json: @book.errors, status: :unprocessable_entity }
38      end
39    end
40  end
end
```

```

41
42 # PATCH/PUT /books/1
43 # PATCH/PUT /books/1.json
44 def update
45   respond_to do |format|
46     if @book.update(book_params)
47       format.html { redirect_to @book, notice: "Book was successfully updated." }
48       format.json { render :show, status: :ok, location: @book }
49     else
50       format.html { render :edit }
51       format.json { render json: @book.errors, status: :unprocessable_entity }
52     end
53   end
54 end
55
56 # DELETE /books/1
57 # DELETE /books/1.json
58 def destroy
59   @book.destroy
60   respond_to do |format|
61     format.html { redirect_to books_url, notice: "Book was successfully destroyed." }
62     format.json { head :no_content }
63   end
64 end
65
66 private
67 # Use callbacks to share common setup or constraints between actions.
68 def set_book
69   @book = Book.find(params[:id])
70 end
71
72 # Only allow a list of trusted parameters through.
73 def book_params
74   params.require(:book).permit(:title, :number_of_pages, :image, :detail)
75 end
76 end

```

7. /app/views/users/show.html.erb

```

1 <h1>プロフィール</h1>
2
3 <p>
4   <%= if @user.avatar.attached? %>
5     <%= @user.avatar %>
6
7   <%= end %>
8 </p>
9
10 <h3>名前</h3>
11 <%= @user.name %>
12
13 <h3>年齢</h3>
14 <%= @user.age %>
15
16 <h3>メールアドレス</h3>
17 <%= @user.email %>

```

8. app/models/user.rb

```
1 class User < ApplicationRecord
2   # Include default devise modules. Others available are:
3   # :confirmable, :lockable, :timeoutable, :trackable and :omniauthable
4   devise :database_authenticatable, :registerable,
5         :recoverable, :rememberable, :validatable
6
7   has_one_attached :avatar
8
9   validates :name, presence: true
10  validates :age, presence: true
11 end
```

9. ./app/views/static_pages/home.html.erb

```
1 <h1>ホームページ</h1>
2 <%= link_to "本一覧をみる", books_path %>
```

10. ./app/views/books/show.html.erb

```
1 <p id="notice"><%= notice %></p>
2
3 <p>
4   <strong>Title:</strong>
5   <%= @book.title %>
6 </p>
7
8 <p>
9   <strong>Number of pages:</strong>
10  <%= @book.number_of_pages %>
11 </p>
12
13 <p>
14   <strong>Detail:</strong>
15   <%= @book.detail %>
16 </p>
17
18 <p>
19   <%= if @book.image.attached? %>
20     <%= image_tag @book.image %>
21   <%= end %>
22 </p>
23
24 <h2>Review Form</h2>
25 <%= form_with(model: @book_review, local: true) do |form| %>
26   <div class= "field">
27     <%= form.text_area :review %>
28     <%= form.hidden_field :book_id, :value => @book.id %>
29   </div>
30   <div class="actions">
31     <%= form.submit %>
32   </div>
33 <%= end %>
34
35 <h2>Review List</h2>
36 <ul>
37   <%= @book.book_review.each do |book_review| %>
38     <li><%= book_review.review %>
39       (<i><%= book_review.user.name if book_review.user %></i></li>
40   <%= end %>
41 </ul>
42
```

Question **26**

Complete

Mark 12.00 out of 12.50

※ 不具合: ホームボタンをクリックしても「ホームページ」が表示されない

・操作:

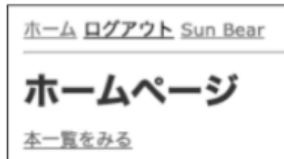


1. 本一覧画面を表示する
2. ホーム ボタンをクリックする

・不具合



・正しい動作



※ 回答

- ・修正するファイル名: _____
- ・その行番号: _____
- ・修正したソースコード: _____
- ・不具合の理由と修正した理由: _____

- ファイル: /config/routes.rb
- 行番号: 4
- コード: root 'static_pages#home'
- 理由: RootはBook_reviewsのパスなので、ホームページが表示されない

Question **27**

Complete

Mark 12.50 out of 12.50

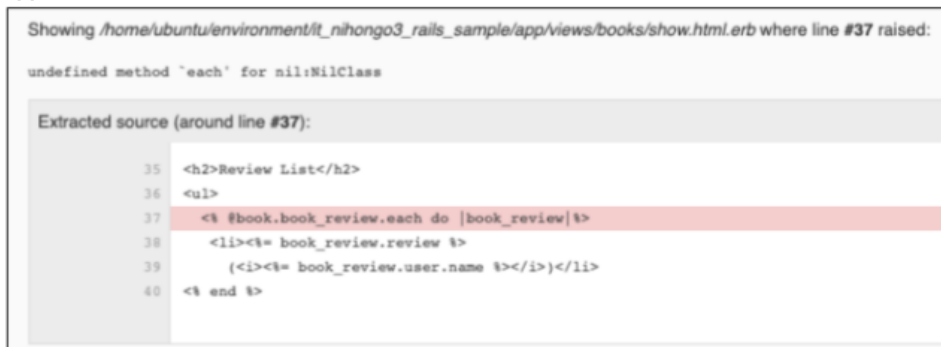
※ 不具合: ボタンをクリックしても本の詳細画面が表示されない

・ 操作:



1. 本一覧画面を表示する
2. Show ボタンをクリックする

・ 不具合



・ 正しい動作



※ 回答

- ・ 修正するファイル名: _____
- ・ その行番号: _____
- ・ 修正したソースコード: _____
- ・ 不具合の理由と修正した理由: _____

- ファイル: /app/models/book.rb
- 行番号: 5
- コード: has_many :book_reviews
- 理由: 一つのBookは多数のレビューがあります

Question **28**

Complete

Mark 0.00 out of 25.00

プレゼンテーションで使うと、聞き手がより興味を持って聞いてくれると思うテクニックを一つあげて、その理由を説明しなさい。
そして、そのテクニックを使った簡単なスクリプトを作りなさい。

字数制限: 300字以上

※ 300字に満たない回答は点数半分（13点）を満点として採点します。

スクリプトを作る時、ジェスチャーなど動作を示すときは（ ）を使って書きなさい。

例: 私が、是非皆さんにお伝えしたいことが（人差し指を1本立てて）一つあります。

