


学習プラットフォームとしての OpenCEAS

An aerial photograph of the Kio University campus, showing a large brick building complex with a central courtyard and a clock tower. The image is partially obscured by a white circular gradient.

畿央大学 教育学習基盤センター 特任助教
宮崎 誠

✉ m.miyazaki@kio.ac.jp

担任者TOPページ



教材作成および登録

授業資料



複合式テスト作成



記号入力式テスト作成



レポート課題作成



アンケート作成



評価記入リスト作成



教材一括更新



教材の授業への割付

教材割付



科目環境設定



科目独自のページ



授業データ管理

複合式テスト管理



記号入力式テスト管理



レポート管理



アンケート管理



評価記入リスト管理



出席管理



連結評価一覧表



授業補助ツール

お知らせ/メール



FAQ回答登録



お知らせ

情報処理演習II (1Ed対象) /AviUtlで... 2018/01/10 18:24:12

情報処理演習II (1Ed対象) /今週の課題について 2017/12/19 18:25:08

情報処理演習II (1Ed対象) /課題の提出について 2017/12/14 15:10:04

情報処理演習II (1Ed対象) /本日12/5特... 2017/12/05 16:19:39

情報処理演習II (1Ed対象) /絵コンテの提出... 2017/11/28 10:04:30

お知らせ一覧

公開FAQ

まだデータは登録されていません

公開FAQ一覧

学生としての試行画面

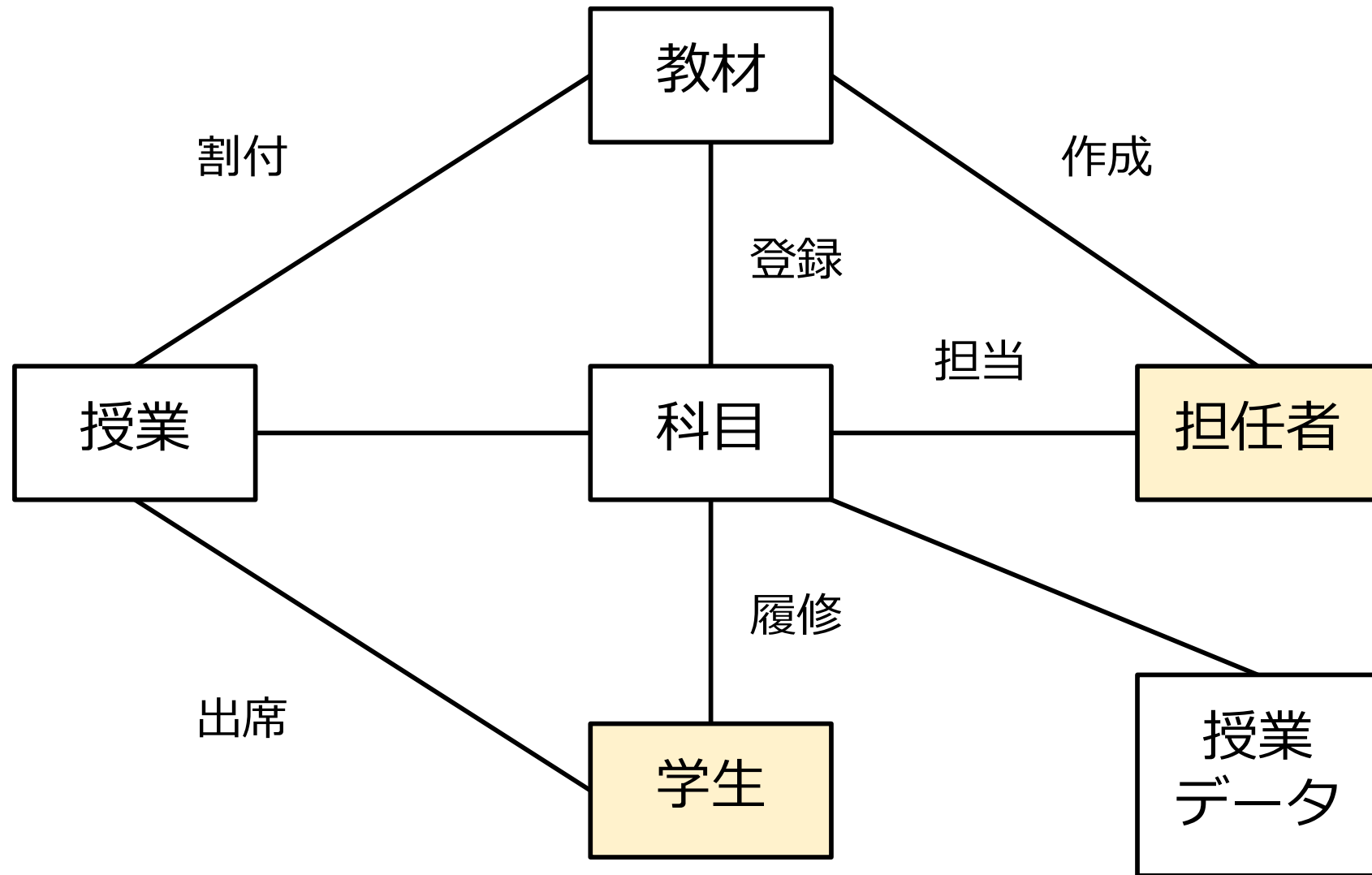
該当数2件

教材の割付確認・授業時の学生画面模擬・出席確認指示等ができます

No.	曜日	時限	年度	学期	科目名称	担任者	系・学科・コース	授業実施画面へ
1	月曜日	1時限	2017	後期	情報処理演習II (1Ed対象)	宮崎 誠		授業実施画面へ
2	火曜日	4時限	2007	春	情報システム演習	夏木正	システムマネジメント	授業実施画面へ

△先頭へ

「授業」を明示的に考慮した分析モデル



基本機能の洗練と使い方の工夫

教務システムからのデータ連携による科目，履修者情報等の自動登録
統合認証システム(CAS)による認証システムの一元化

- 教員 – 担任者TOPページ
 - 担当科目の一覧
 - 授業実施プロセス（準備→授業→評価）毎のメニュー配置
 - 教材やテストの「作成」→「割付」→「採点，管理」
 - レポートの「作成」→「割付」→「採点，管理」
 - お知らせ，FAQなど
- 学生 – 学生TOPページ
 - 受講科目の一覧
 - レポートやテストの一覧画面へのリンク
 - お知らせ，FAQなど

操作によるグループ化(担任者TOPページ)

教材作成および登録

- 授業資料
- 教材の作成/登録/割付**
- 複合式テスト作成
- レポート課題作成
- 教材作成/登録
- アンケート作成/登録
- レポート作成/登録
- アンケート作成/登録
- 教材一括登録
- 授業への割付
- 教材の授業への割付
- 教材割付
- 科目環境設定
- 科目独自のページ

授業データ管理

- 複合式テスト採点/管理
- 記号入力テスト採点/管理
- アンケート結果/管理
- レポート管理
- アンケート管理
- 評価記入リスト管理
- 出席管理
- 連結評価一覧表

都度対応ツール

- お知らせ
- FAQ

お知らせ

- 情報処理演習II (1Ed対象) /AviUtlで... 2018/01/10 18:24:12
- 情報処理演習II (1Ed対象) /今週の課題について 2017/12/19 18:25:08
- 情報処理演習II (1Ed対象) /課題の提出について 2017/12/14 15:10:04
- 情報処理演習II (1Ed対象) /本日12/5特... 2017/12/05 16:19:39
- 情報処理演習II (1Ed対象) /絵コンテの提出... 2017/11/28 10:04:30

公開FAQ

まだデータは登録されていません

[公開FAQ一覧](#)

[お知らせ一覧](#)

学生としての試行画面 該当数2件

教材の割付確認・授業時の学生画面模擬・出席確認指示等ができます

授業実施

No.	曜日	時限	2017	後期	科目名称	担任者	系・学科・コース	授業実施画面へ
1	月曜日	1時限	2017	後期	情報処理演習II (1Ed対象)	宮崎 誠		授業実施画面へ
2	火曜日	4時限	2017	後期	システム管理	夏木正	システムマネジメント	授業実施画面へ

[先頭へ](#)

都度対応の情報表示

授業実施

- 担当科目一覧
- 授業実施画面へ遷移

授業実施プロセス別操作カテゴリの配置

OpenCEAS 学生画面

OpenCEAS
(PC表示)

OpenCEAS
(スマホ表示)

CEAS 学生TOPページ

レポート確認 公開科目一覧 操作マニュアル 0000002畿央 二郎

ユーザ情報

お知らせ

- 1P畿央 太郎 IV/テスト 2015/10/02 12:01:59
- 1P畿央 太郎 IV/TEST 2015/10/02 09:42:59
- 1P畿央 太郎 IV/Windows10での... 2015/10/01 15:56:42
- 1P畿央 太郎 IV/TEST2015091... 2015/09/15 18:07:44
- 1P畿央 太郎/TEST20150915 2015/09/15 18:06:59

お知らせ一覧

公開FAQ

まだデータは登録されていません

公開FAQ一覧

CEAS

学生TOPページ

ユーザ情報

表示言語切替

日本語 変更

お知らせ

公開FAQ

授業画面の表示 **該当数 1件**

1 1P畿央 太郎

授業実施画面へ

テスト結果へ

先頭へ

CEAS 学習進捗-分析実行 --テスト結果一覧--

TOPページ / 科目別進捗状況

1P畿央 太郎 0000002畿央 二郎

受験状況 詳細データ

No.	テスト方式	タイトル	受験回数	点数/満点	順位	平均点	最高得点	確認
1	複合式テスト	2015年度第1回事前テスト	0	-	-	-	-	-
2	記号入力式テスト	TEST	0	-	-	-	-	-
3	記号入力式テスト	記号TEST	1	0/100	1	0	0	確認する
4	記号入力式テスト	人体の構造と機能および疾病の成り立ち-1	3	0/100	1	0	0	確認する
5	複合式テスト	人体の構造と機能および疾病の成り立ち-1	2	18/100	1	18	18	確認する
6	複合式テスト	人体の構造と機能および疾病の成り立ち-2	2	14/100	1	14	14	確認する

テスト受験 進捗データ

全テスト数	6
受験済みテスト数	4
受験率	66 %

なぜ教員が自発的に使い出せるのか

1. CEASが想定する**授業実施プロセスモデル**と、**教員の授業実施の活動フロー**が一致している

(「**準備** → **授業** → **評価**」の毎週のサイクルと成績評価)

担任者、学生の教育/学習に含まれる諸活動を毎回の「授業」の実施を軸としたフローとしてとらえている

=> 直感的に挙動を想定でき、操作が分かり易い

2. ユーザーインターフェイスがユーザビリティの条件をかなり満たしている

- 学習しやすさ
- 効率 (タスク実行効率)
- 記憶しやすさ
- エラー (数、深刻さ、回復容易) *Jakob Nielsen, Usability Engineering, Academic Press, 1994*
- 満足 (使うのが楽しいか) *Academic Press, 1994*

=> CEASのユーザーインターフェイスが持つ特徴を一般化し、
「**授業支援型ユーザーインターフェイス**」と定義

Ruby on Rails の基本理念

Railsの哲学には、以下の2つの主要な基本理念があります。

- **同じことを繰り返すな (Don't Repeat Yourself: DRY):** DRYはソフトウェア開発上の原則であり、「システムを構成する知識のあらゆる部品は、常に単一であり、明確であり、信頼できる形で表現されていなければならない」というものです。同じコードを繰り返し書くことを徹底的に避けることで、コードが保守しやすくなり、容易に拡張できるようになり、そして何よりバグを減らすことができます。
- **設定より規約が優先される (Convention Over Configuration):** Railsでは、Webアプリケーションで行われるさまざまなことを実現するための最善の方法を明確に思い描いており、Webアプリケーションの各種設定についても従来の経験や慣習を元に、それらのデフォルト値を定めています。このようにある種独断でデフォルト値が決まっているおかげで、開発者の意見をすべて取り入れようとした自由過ぎるWebアプリケーションのように、開発者が延々と設定ファイルを設定して回らずに済みます。

DRY(Don't Repeat Yourself)原則

Viewの比較

	ファイル数	総行数
CEAS3	202	32,678
OpenCEAS	294	29,910

↑
45.5%増

↑
-8.5%減

erbファイル数は増えたが、「テンプレート」を工夫することにより、内部は記述が簡潔に

総行数はあまり変化していない

CoC(Convention over Configuration)

「設定よりも規約」とは、「慎重に設計された規約 (Convention) に従うことにより、設定 (Configuration) を不要にする (あるいは軽減する)」ということ

Wikipedia Ruby on Rails, https://ja.wikipedia.org/wiki/Ruby_on_Rails

規約による関連付けの例(ActionとView)

controllers/announcements_controller.rb

```
class AnnouncementsController < ApplicationController
```

```
  def index
```

```
  ...
```

```
  end
```

```
  def show
```

```
  ...
```

```
  end
```

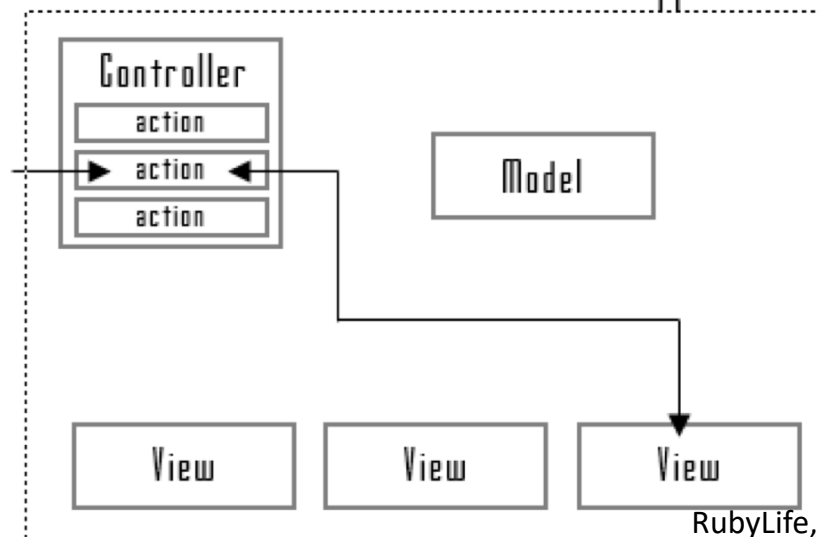
```
end
```

views/announcement/

index.html.erb

views/announcement/

show.html.erb



CoC(Convention over Configuration)

「設定よりも規約」とは、「慎重に設計された規約 (Convention) に従うことにより、設定 (Configuration) を不要にする (あるいは軽減する)」ということ

Wikipedia Ruby on Rails, https://ja.wikipedia.org/wiki/Ruby_on_Rails

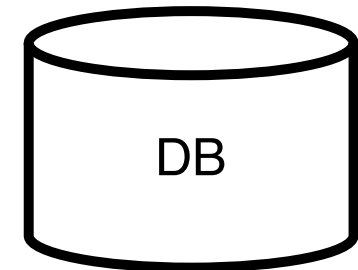
規約による関連付けの例(Modelとテーブル)

models/announcement.rb

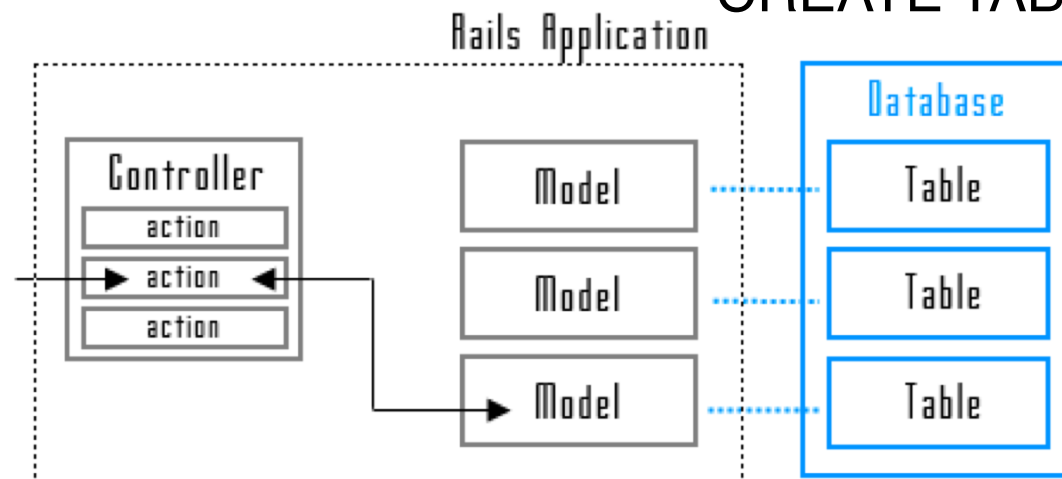
```
class Announcement < ActiveRecord::Base
```

```
...
```

```
end
```



CREATE TABLE announcements ...



RubyLife, <https://www.rubylife.jp/rails/ini/index7.html>

REST(REpresentational State Transfer)原則

```
miyazaki — -bash — 123x37

$ rake routes

      Prefix Verb   URI Pattern                                     Controller#Action
  user_password POST  /users/password(.:format)                   devise/passwords#create
  new_user_password GET    /users/password/new(.:format)               devise/passwords#new
  edit_user_password GET    /users/password/edit(.:format)              devise/passwords#edit
  PATCH      /users/password(.:format)                   devise/passwords#update
  PUT        /users/password(.:format)                   devise/passwords#update
  cancel_user_registration GET    /users/cancel(.:format)                     devise/registrations#cancel
  user_registration POST   /users(.:format)                             devise/registrations#create
  new_user_registration GET    /users/sign_up(.:format)                    devise/registrations#new
  edit_user_registration GET    /users/edit(.:format)                       devise/registrations#edit

edit_user_registration GET    /users/edit(.:format)                       devise/registrations#edit
  PATCH      /users(.:format)                             devise/registrations#update
  PUT        /users(.:format)                             devise/registrations#update
  DELETE     /users(.:format)                             devise/registrations#destroy

      Prefix Verb   URI Pattern                                     Controller#Action
  users GET    /users(.:format)                             users#index
  system_general_announcements GET    /general_announcements/system(.:format)     general_announcements#system
  general_announcements GET    /general_announcements(.:format)            general_announcements#index
  general_announcement GET    /general_announcements/:id(.:format)         general_announcements#show
  announcements GET    /announcements(.:format)                    announcements#index
  announcement GET    /announcements/:id(.:format)                 announcements#show
  faq_answers GET    /faq_answers(.:format)                       faq_answers#index
  faq_answer GET    /faq_answers/:id(.:format)                   faq_answers#show
  class_sessions GET    /class_sessions/:course_id(.:format)         class_sessions#index
  class_session GET    /class_sessions/:course_id/show(.:format)    class_sessions#show
  GET        /class_sessions/:course_id/announcement(.:format)
  class_sessions#announcement
  GET        /class_sessions/:course_id/faq(.:format)     class_sessions#faq
  GET        /class_sessions/:course_id/specific_page(.:format)
  class_sessions#specific_page
  collect_attendance GET    /class_sessions/:course_id/collect_attendance(.:format)
  class_sessions#collect_attendance
  POST      /class_sessions/:course_id/start_collect_attendance(.:format)
  class_sessions#start_collect_attendance
```

リソース識別子(URI)

コントローラー、
アクション

edit_user_registration	GET	/users/edit(.:format)	devise/registrations#edit
	PATCH	/users(.:format)	devise/registrations#update
	PUT	/users(.:format)	devise/registrations#update
	DELETE	/users(.:format)	devise/registrations#destroy

リクエスト

学習プラットフォーム

- オールインワン型からモジュール型へ
 - Web API…データDRIVENの相互運用, 統合
 - メリット
 - LMSの世界で完結するのでユーザインターフェイスやWebデザインが変わらない
 - デメリット
 - データ以外の画面遷移等一部ロジック層やプレゼンテーション層の実装が必要
 - データモデルが変更になると影響を受ける
 - LTI…ツールDRIVENの相互運用, 統合
 - メリット
 - LTIプロバイダLAUNCHの設定だけでツールを利用可能
 - デメリット
 - LTIプロバイダの世界で利用するのでユーザインターフェイスやWebデザインが変わる

NGDLEの5次元(“Dimensions of the NGDLE”)

- **Interoperability and Integration 相互運用性と統合**
ツールを統合し, コンテンツや学習データを交換できる
- **Personalization パーソナル化**
一人ひとりに適した学習環境と適応学習 (学習過程の最適化)
- **Analytics, Advising and Learning Assessment**
解析・助言・学習測定
学習データ解析を利用した新たな学習者支援・学習評価
- **Collaboration コラボレーション**
協調学習
- **Accessibility and Universal Design**
アクセシビリティ・ユニバーサルデザイン
だれもが, すぐに利用できる