



likeLion
HACK YOUR LIFE



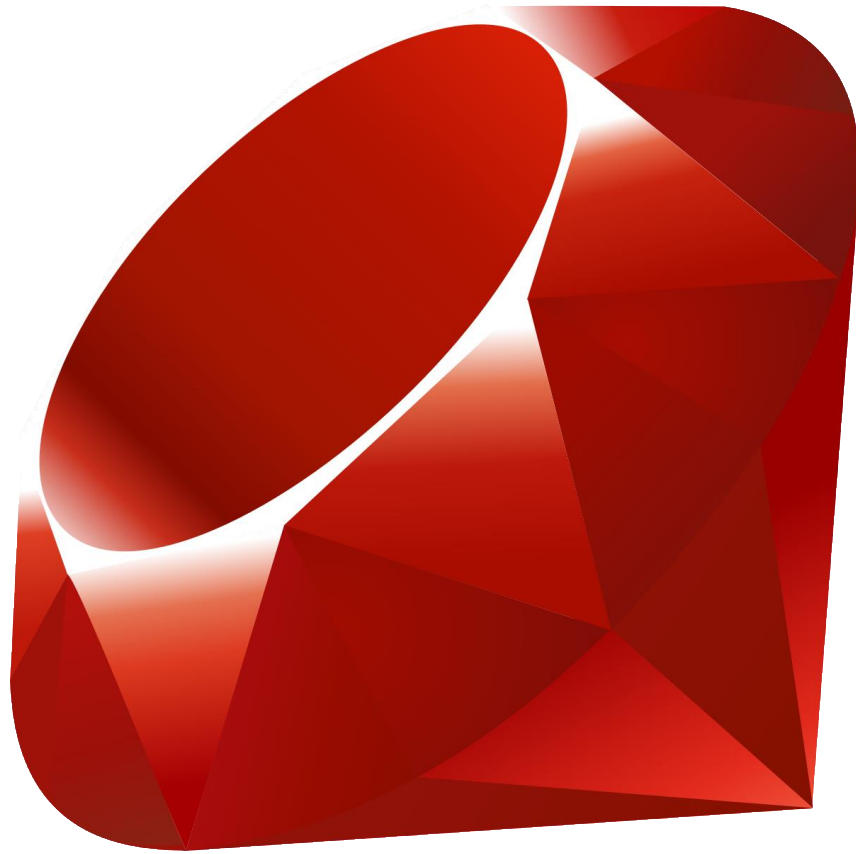
likeLion
멋쟁이 사자처럼




MVC, CRUD



▀ Ruby on Rails ▾



 Ruby on Rails,

Ruby

객체 지향 프로그래밍 언어

Object <대상>

Ruby on Rails,

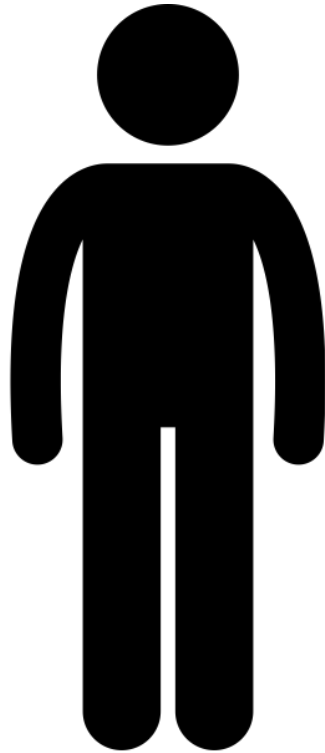
Object <대상>

대상 = 정보 + 행위

Object = 변수 + 함수

Ruby on Rails,

person



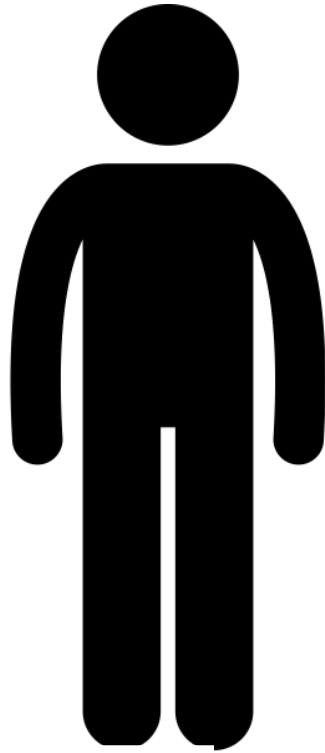
Ruby on Rails,
변수

@name = 고우영

@sex = 남

@age = 26

@hobby : 누워있기



Ruby on Rails,
메서드 <함수>



person.walk



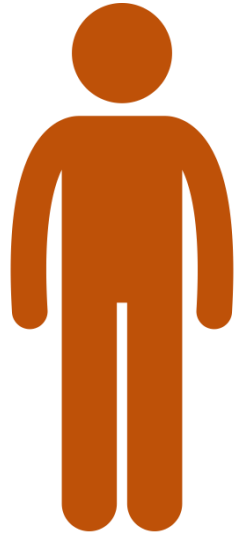
person.run



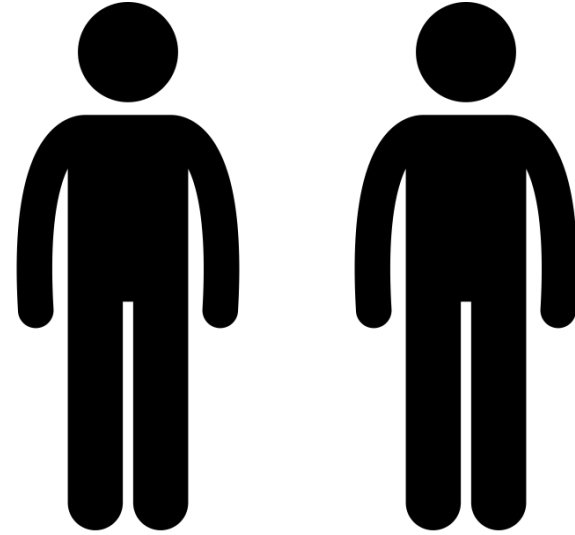
person.swim

Ruby on Rails,

Object



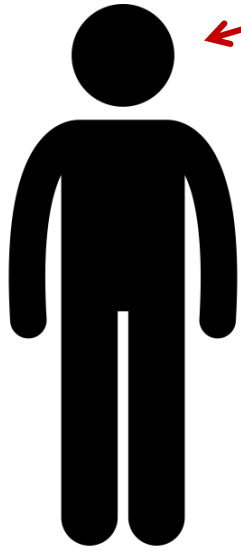
클래스(class)
반, 분류 하나의 기준



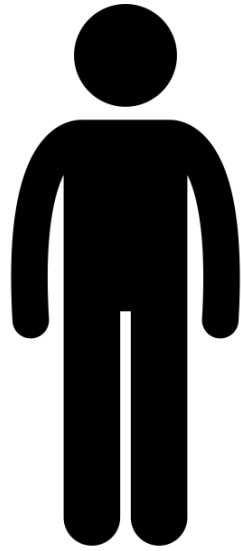
인스턴스(instance)
구체적인 예

Ruby on Rails,

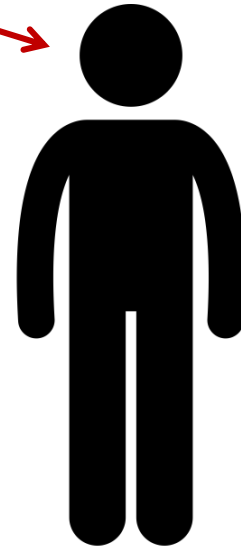
class(하나의 기준) = person



전현무/42/남



한혜진/36/여

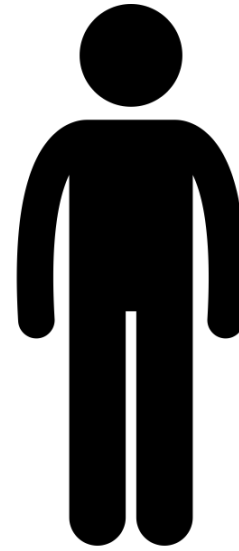
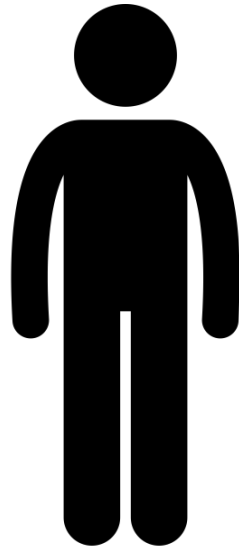
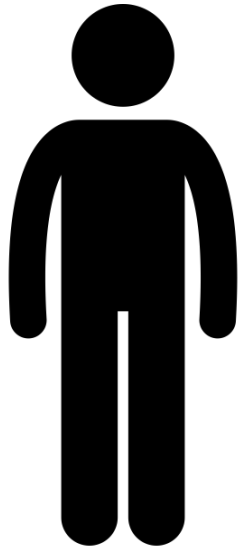


박나래/34/여

걷기, 달리기, 수영하기

Ruby on Rails,

person



각자 다른 정보 / 변수

같은 행동 / 메서드(함수)

그리고 다른 결과!



전현무.walk

13초



한혜진.walk

12초



박나래.walk


11초

Ruby on Rails



Ruby on rails

루비로 작성된 MVC패턴 기반의 오픈 소스 웹 프레임워크

 Ruby on Rails,

MVC

레일즈에 적용되는 디자인 패턴

Model

M

데이터를 관리

View

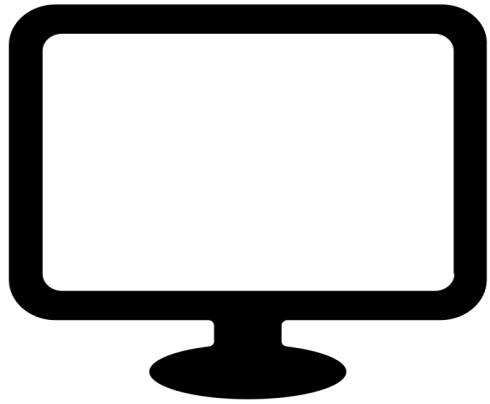
V

사용자가 보는 화면

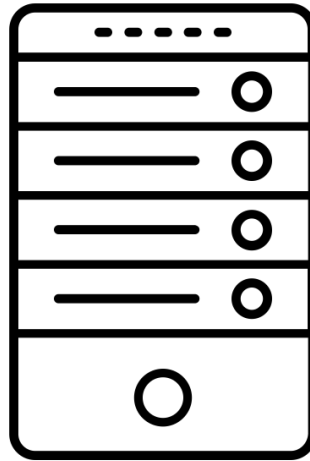
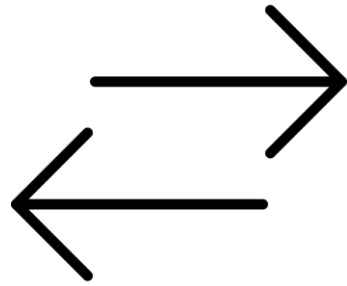
Controller

C

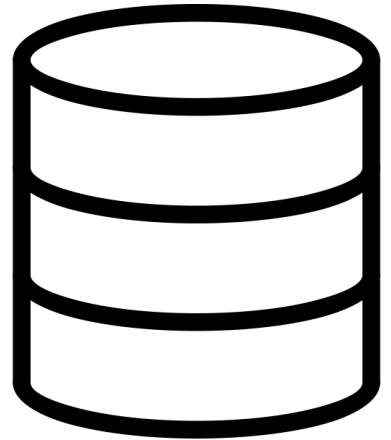
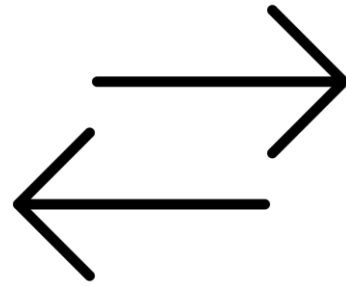
중간 관리자



클라이언트



서버

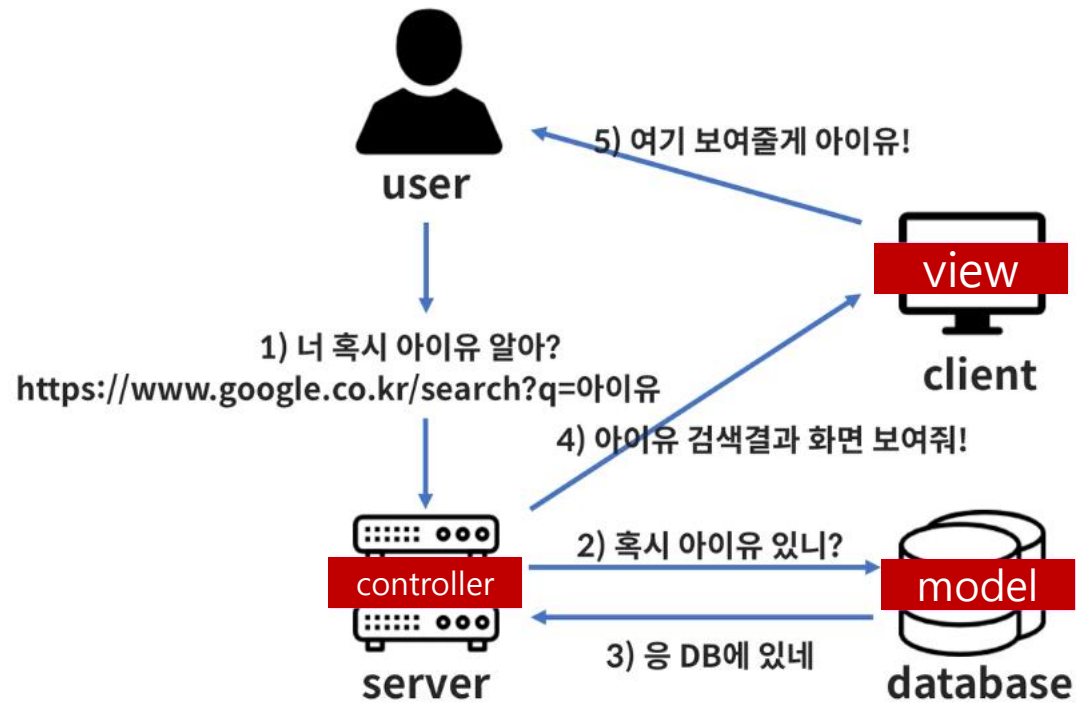


데이터베이스



Ruby on Rails,

MVC



구글에 아이유를 검색했을 때 MVC 패턴

V : View

결과를 출력

C : Controller

컨트롤러 : 액션들의 집합

=> 액션 : 실질적인 요청을 처리

C : Controller

Controller 이름과 같은 views 디렉토리에 Action 이름과 같은 View 파일(html.erb)이 있다면 그걸 보여준다.

액션 == 뷰

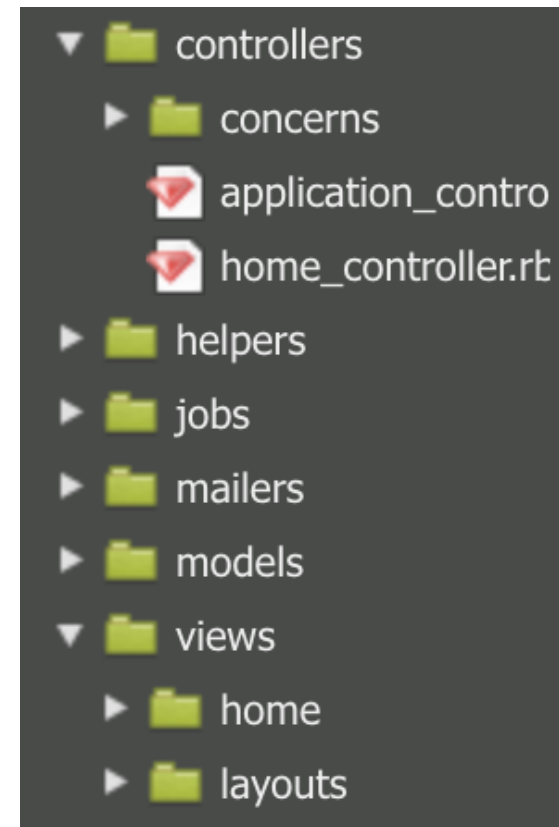
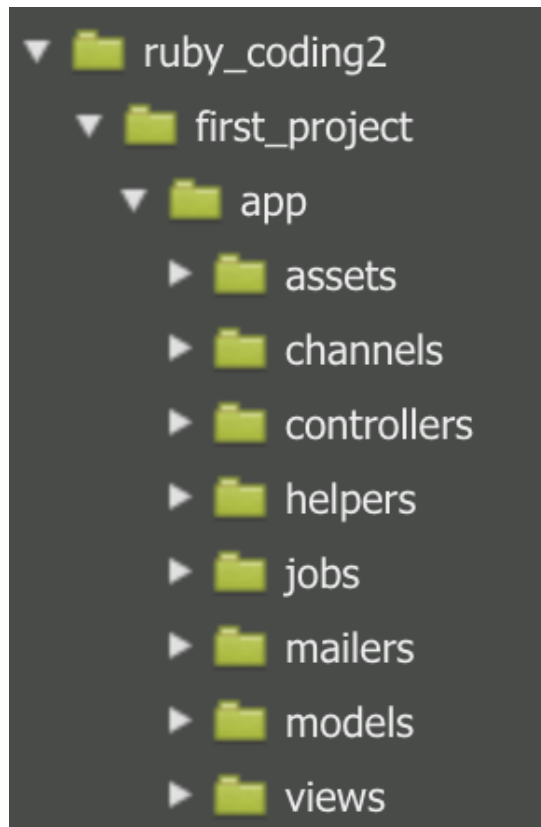
컨트롤러 생성
rails g controller 이름 액션1 액
션2

rails d controller 이름 : 컨트롤러 삭제

Ruby on Rails

C : Controller

```
songyi:~/workspace/first_project $ rails g controller home
Running via Spring preloader in process 14660
  create  app/controllers/home_controller.rb
  invoke  erb
  create  app/views/home
  invoke  test_unit
  create  test/controllers/home_controller_test.rb
  invoke  helper
  create  app/helpers/home_helper.rb
```

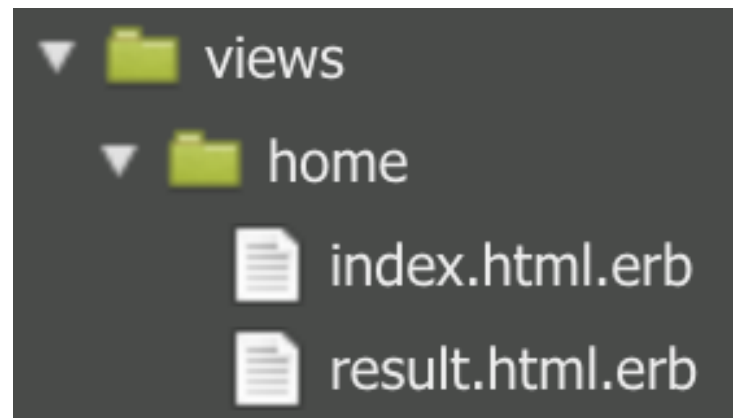


Ruby on Rails

C : Controller

```
1 class HomeController <
2   def index
3     #정 보 를 처 리
4   end
5
6   def result
7   end
8 end
```

action 만들기



view(결과) 만들기

Route

길을 안내하는 표지판

어떠한 컨트롤러와 액션으로 갈지 정해준다.

Ruby on Rails

config/routes.rb

어떤 주문이 들어오면 이런 방식으로 저런 동작을 해라

```
Rails.application.routes.draw do
  root 'home#index'

  get 'home/index'
  #get 'home/index' => 'home#index'
  get 'home/new'
  #get 'home/new' => 'home#new'
  get 'home/create'
  #get 'home/create' => 'home#create'
end
```

사용자가 초록색 박스의 방식으로 빨간색 박스의 URL로
요청을 하면 컨트롤러가 파란색 박스의 액션을 처리함

```
root 'posts#index'  
get 'posts/index'
```

- root는 가장 처음 보이는 메인 페이지의 경로를 설정
'/' => 'posts#index'
- get은 HTTP Request 방식으로 전송할 정보를 URL에
담아서 보내는 것
- post는 U | 보낸다.

M : Model

데이터베이스와 대응되는 개념

Ruby on Rails,

M : Model

루비 문법을 이용하여 모델을 사용하면
SQL을 사용하여 DB에 접근 하듯 이용 가
능

Ruby on Rails,

M : Model

SQL(Structured Query Language)

RDBMS의 데이터를 관리하기 위해 설계된 특수 목적
언어

Ruby on Rails,

M : Model

0. Model 생성 및 Table 이름 정하기.

Avengers

M : Model

1. Table 구조 잡기 (Column 구성)

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)

Ruby on Rails,

M : Model

2. Table 실제 생성

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)

M : Model

3. 데이터 1줄 생성(Row 추가)

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)

Ruby on Rails,

M : Model

4. 데이터 채워 넣기

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)
	아이언맨	남성	45	펄프

Ruby on Rails,

M : Model

5. 데이터 저장

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)
1	아이언맨	남성	45	펄프

Ruby on Rails,

M : Model

3 - 4 - 5 반복

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)
1	아이언맨	남성	45	펄프

Ruby on Rails,

M : Model

3 - 4 - 5 반복

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)
1	아이언맨	남성	45	펄프
	토르	남성	1500	천둥

Ruby on Rails,

M : Model

3 - 4 - 5 반복

Avengers

	Name(string)	Sex(string)	Age(integer)	Ability(string)
1	아이언맨	남성	45	펄프
2	토르	남성	1500	천둥

Ruby on Rails,

M : Model

0. Model 생성 및 Table 이름 정하기.

이름 = avenger

1. Table 구조 잡기 (Column 구성)

구성 = Name, Gender, Age, Ability

Ruby on Rails,

M : Model

Rails **g** model **avenger(단수)** name:string
gender:string age:string ability:string

Ruby on Rails,

M : Model

2. Table 실제 생성

```
rails db:migrate
```

```
rake db:migrate
```

▸ Ruby on Rails,

M : Model

3. 데이터 1줄 생성(Row 추가)

```
a = Avenger.new
```

M : Model

4. 데이터 채워 넣기

```
a = Avenger.new
```

```
a.name = '___'
```

```
a.age = ___
```

```
...
```

▮ Ruby on Rails,

M : Model

5. 데이터 저장

`a.save`

Ruby on Rails,

M : Model

조회

Avenger.all

Avenger.find 1(ID값)

수정

a = Avenger.find 1

▸ Ruby on Rails,

M : Model

Column 구성의 수정은?

`rails db:drop`

후 수정, but 정보 모두 날아가..

Ruby on Rails,

CRUD

게시판 기능

CRUD 란?

데이터를 효과적으로 다루기 위한 패턴

Create

Read

Update

Delete

C

R

U

D

데이터 생성

데이터 읽기

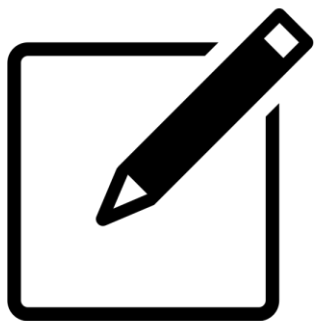
데이터 갱신

데이터 삭제

CRUD를 사용해 게시판을 만든다면?

글 작성, 보기, 변경, 삭제

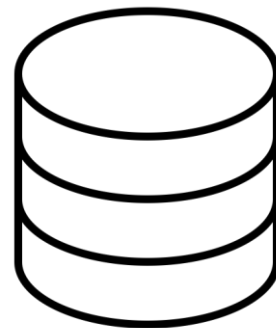
Create(생성)



글 작성

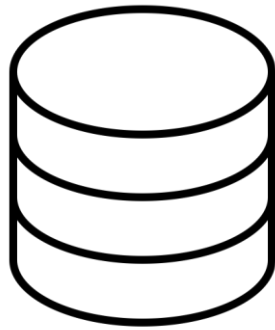


저장

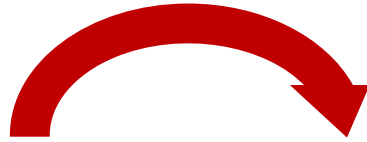


모델(DB)

Read(읽기)

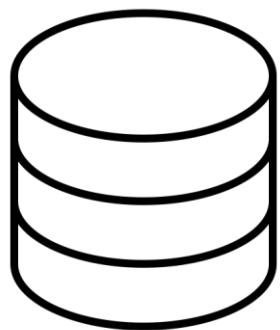


모델(DB)

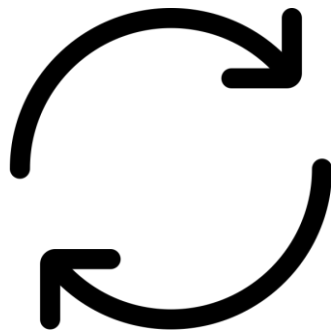


작성 글 읽기

Update(갱신)

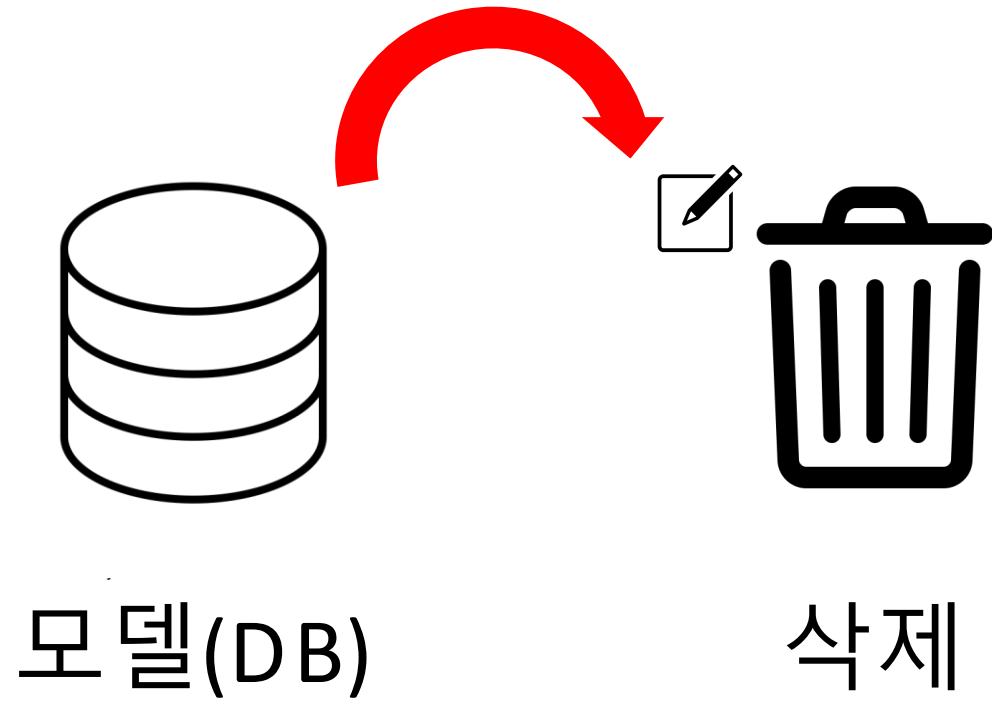


모델(DB)



새롭게 작성된 글

Delete(삭제)





배운 내용으로 CRUD를 구현한다면?

View & Controller

입력한 정보를
테이블에
넣어주는 기능

+

Model

게시판에 필요한
제목과 내용을
저장할 테이블(표)

VC

컨트롤러 만들기

액션 만들기

라우트 작성하기

뷰 .html.erb 만들기

+

M

모델 만들기

스키마 작성하기

마이그레이션 하기

저장

VC

rails g **controller**

controller#action

get 'home/index'

home.**html.erb**

+

M

rails g **model**

Model.string :title

rails **db:migrate**

Model.save

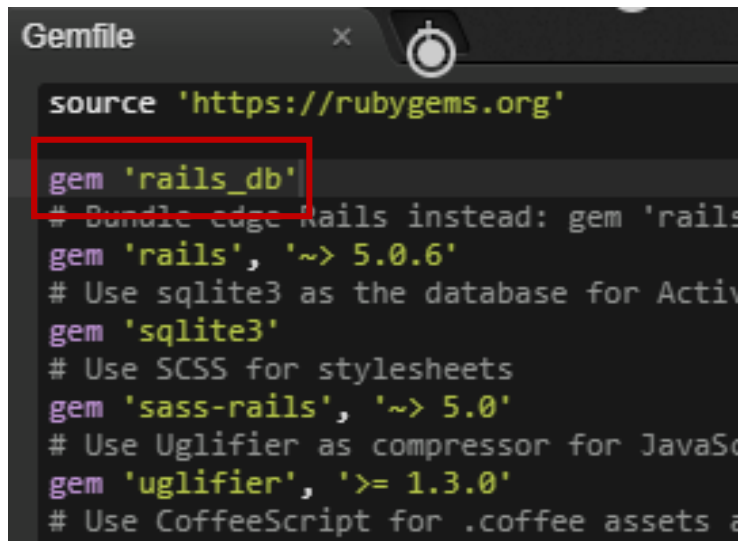
 Ruby on Rails,

CRUD

 Ruby on Rails,

Create

gem 'rails_db'



```
Gemfile
source 'https://rubygems.org'
gem 'rails_db'
# Bundle edge Rails instead: gem 'rails'
gem 'rails', '~> 5.0.6'
# Use sqlite3 as the database for Active Record
gem 'sqlite3'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5.0'
# Use Uglifier as compressor for JavaScript
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .coffee assets and templates
```

***rails_db** :데이터베이스를 쉽게 다루기 위한 프로그램

***gem** : 루비 프로그램 관리자

***bundle install** 명령어로 설치

rails g model 모델명

Model 생성

```
kkodu:~/workspace $ rails g model Note
Running via Spring preloader in process 13067
  invoke  active_record
  create  db/migrate/20180502180610_create_notes.rb
  create  app/models/application_record.rb
  create  app/models/note.rb
  invoke  test_unit
  create  test/models/note_test.rb
  create  test/fixtures/notes.yml
kkodu:~/workspace $
```

생성된 내용은 **db/migrate** 폴더에
날짜_create_테이블명.rb 파일에 있음

- 테이블의 모양을 정의
- 기본적으로 id 자동 생성, 주민등록번호 역할
- 게시물 수정, 삭제 시 꼭 필요한 요소
- 구성만 된 상태이며 확정된 상태는 아님

table.type :컬럼명(속성)

Model에 여러 속성을 부여

```
class CreateNotes < ActiveRecord::Migration[5.0]
  def change
    create_table :notes do |t|
      t.string :title
      t.text :content
      t.timestamps
    end
  end
end
```

타입

- Integer – 정수
- String – 문자열
- Text – 문자열 확장

컬럼 - 데이터의 속성

rails db:migrate

수정 내용을 rails_db에 적용 -> 테이블 확정

```
kkodu:~/workspace $ rails db:migrate
== 20180502180610 CreateNotes: migrating =====
-- create_table(:notes)
   -> 0.0012s
== 20180502180610 CreateNotes: migrated (0.0012s) =====

kkodu:~/workspace $
```

* rails db:drop - 테이블 취소(migrate가 선행되어 있어야 함)

콘솔에서 Model에 데이터 저장

콘솔창 열기

```
kkodu:~/workspace $ rails c
Running via Spring preloader in process 13135
Loading development environment (Rails 5.0.7)
2.3.4 :001 > note = Note.new
=> #<Note id: nil, title: nil, content: nil, created_at: nil, updated_at: nil>
2.3.4 :002 > note.title = "오늘할일"
=> "오늘할일"
2.3.4 :003 > note.content = "멋사강의, 수업"
=> "멋사강의, 수업"
2.3.4 :004 > note.save
(0.2ms) begin transaction
SQL (0.5ms) INSERT INTO "notes" ("title", "content", "created_at", "updated_at") VALUES (?, ?, ?, ?)
", "2018-05-02 18:11:06.737491"]])
(11.8ms) commit transaction
=> true
2.3.4 :005 > exit
kkodu:~/workspace $
```

새로운 테이블 행 생성
(하나의 데이터)

실제 rails_db에 쿼리문을 실행하여 만들어진 데이터를 추가

rails_db 저장 확인

Table: notes

Expand

Records

Schema

SQL Query

Export

Actions	Id	Title	Content
<div> <div></div> <div></div> </div>	1	오늘할일	멋사강의, 수업

Total: 1 record

SQL

rails g controller 컨트롤러명(테이블명)

```
kkodu:~/workspace $ rails g controller Notes
Running via Spring preloader in process 14056
  create  app/controllers/notes_controller.rb
  invoke  erb
  create  app/views/notes
  invoke  test_unit
  create  test/controllers/notes_controller_test.rb
  invoke  helper
  create  app/helpers/notes_helper.rb
  invoke  test_unit
  invoke  assets
  invoke  coffee
  create  app/assets/javascripts/notes.coffee
  invoke  scss
  create  app/assets/stylesheets/notes.scss
kkodu:~/workspace $
```

new와 create 액션 생성

```
class NotesController < ApplicationController
  # 입력폼(New) => notes#new
  def new

  end

  # 생성(Create) => notes#create
  def create

  end
end
```

notes#new

- 게시물 내용을 입력받을 액션

notes#create

- 입력된 데이터를 전달받아서 테이블에 저장하는 액션

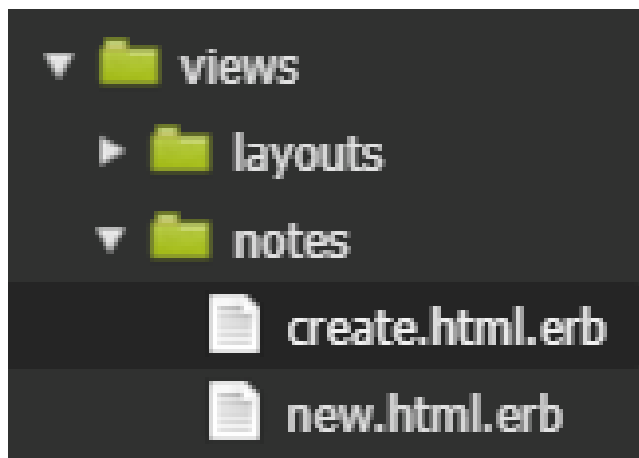
route 설정

```
Rails.application.routes.draw do
  # root
  get '/' => 'notes#new'

  # get 'notes/create'
  get 'notes/create' => 'notes#create'
end
```

- 입력 페이지는 **new**
- 저장 페이지는 **create**

view를 위한 .html.erb 파일 생성



views/notes 폴더에 위치

new.html.erb

```
<h1>제목과 내용을 입력해주세요!</h1>

<form action="/notes/create">
  <input type="text" name="title">
  <input type="text" name="content">
  <input type="submit" value="create!!">
</form>
```

제목과 내용을 입력해주세요!

뭐하지.. 영화보기 create!!

Action - 실행할 액션 경로를 설정, 데이터는 **params** 변수에 저장

Name - 사용자가 입력한 데이터가 담기는 변수명

Value - 사용자가 입력한 데이터

Submit - action 경로로 전송

create.html.erb

- 전달받은 데이터가 **params**에 담겨
사용된다.

```
# 생성(Create) => notes#create
def create
  # puts params 확인
  note = Note.new
  note.title = params[:title]
  note.content = params[:content]
  note.save

  redirect_to '/'
end
```

- **note.new**로 테이블에 데이터 한 줄
추가
- **params**에 담겨온 데이터를 **note**
변수에 할당
- **note.save**로 만들어진 데이터를 테
이블에 저장


≡ 이용해 경로를 재설

Ruby on Rails

create.html.erb

```
Started GET "/notes/create?title=%EB%AD%90%ED%95%98%EC%A7%80..&content=%EC%98%81%ED%99%94%EB%B3%B4%EA%B8
Cannot render console from 61.32.245.1! Allowed networks: 127.0.0.1, ::1, 127.0.0.0/127.255.255.255
Processing by NotesController#create as HTML
Parameters: {"title"=>"뭐하지..", "content"=>"영화보기"}
(0.1ms) begin transaction
SQL (0.3ms) INSERT INTO "notes" ("title", "content", "created_at", "updated_at") VALUES (?, ?, ?, ?)
8-05-02 18:24:42.218648"]
(11.9ms) commit transaction
Redirected to https://cr-practice-kkodu.c9users.io/
Completed 302 Found in 23ms (ActiveRecord: 13.1ms)

Started GET "/" for 61.32.245.1 at 2018-05-02 18:24:42 +0000
```

 Ruby on Rails,

Read

Model.all -> 전체글 조회

```
# New  
def new  
  @posts = Note.all  
end
```

*@posts - posts라는 변수에 모든 테이블 데이터 저장

*@ - .html.erb에서 변수 사용 가능

new.html.erb

```
<h1>글 입력</h1>

<form action="/notes/create">
  <input type="text" name="title">
  <input type="text" name="content">
  <input type="submit">
</form>

<br><hr><br>

<h2>내가 쓴 글</h2>
<% @posts.each do |post| %>
<h3>ID : <%=post.id%> 제목: <%=post.title%></h3>
<% end %>
```

* `<% %>` : ruby 코드 사용 가능

* `<%= %>` : 사용자에게 보여줄 때는 =를 붙여줘야 함

* `.each end` : 한줄 한줄 씩 작업

* `post` : posts 테이블의 각 행

전체글 확인

글 입력

내가 쓴 글

ID : 1 제목: [asd](#)

ID : 2 제목: [abc](#)

ID : 3 제목: [asdf](#)

a 태그 + 경로/id

url에 게시물 id 값을 함께 params에 담아서 보낸다

```
<a href="/notes/read/<%=post.id%>">
```

Route 설정

```
Rails.application.routes.draw do
  get '/' => 'notes#new'
  get 'notes/create' => 'notes#create'

  get 'notes/read/:id' => 'notes#read'
end
```

특정값을 보낼 때는 경로(/) 뒤에
:를 사용한다

: + params명

Model.find id - 하나의 데이터 조회

```
# Read
def read
  @note = Note.find params[:id]
end
```

*route를 통해 넘겨준 id 파라미터 값을 params[:id]로 받는다.

*Note.find로 특정 아이디의 데이터 조회

read.html.erb

```
<h1><%= @note.id %>번 째 글입니다.</h1>
<hr>
<h3>제목 : <%= @note.title %></h3>
<h3>내용 : <%= @note.content %></h3>
```

Ruby on Rails,

게시물 확인

← → ↻ 🔒 안전함 | <https://ruby-crud-kkodu.c9users.io/notes/read/3>

3번 째 글입니다.

제목 : asdf

내용 : asdfafafa