

건 대 멋 사 6 기

멋 사 5 기 이 현 경

VC 패턴 이해하기

contents

Unit 01 | 로컬에서 Rails 시작하기

Unit 02 | VC 이해하기

Unit 03 | Challenge

Unit 01 | 로컬에서 Rails 시작하기

Vagrant 환경에서 rails 프로젝트 시작하기

vagrant up : 가상머신 부팅

vagrant ssh : 가상환경 접속

```
USER@LG-PC ~/Desktop/vagrant
$ vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

21 packages can be updated.
0 updates are security updates.

Last login: Mon Apr  2 10:07:27 2018 from 10.0.2.2
vagrant@ubuntu-xenial:~$
```

Unit 01 | 로컬에서 Rails 시작하기

Vagrant 환경에서 rails 프로젝트 시작하기

cd ../../ : 최상위 디렉토리로 이동

```
vagrant@ubuntu-xenial:/$ ls
bin    etc      lib      media   proc    sbin    sys    vagrant
boot  home    lib64    mnt     root    snap    tmp    var
dev    initrd.img  lost+found  opt     run     srv     usr    vmlinuz
```

cd vagrant : 내 로컬의 공유폴더로 이동

```
vagrant@ubuntu-xenial:/vagrant$ ls
hi.txt      quiz.rb      ubuntu-xenial-16.04-cloudimg-console.log
horyeong ruby  quiz.zip    Vagrantfile
prl         train_data.json  Vagrantfile.zip
```

Unit 01 | 로컬에서 Rails 시작하기

Rails Project 생성

rails new [프로젝트명]

```
vagrant@ubuntu-xenial:/vagrant$ rails new w3_prac1
create
create README.md
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/config/manifest.js
create app/assets/javascripts/application.js
create app/assets/javascripts/cable.js
create app/assets/stylesheets/application.css
create app/channels/application_cable/channel.rb
create app/channels/application_cable/connection.rb
create app/controllers/application_controller.rb
create app/helpers/application_helper.rb
create app/jobs/application_job.rb
create app/mailers/application_mailer.rb
create app/models/application_record.rb
create app/views/layouts/application.html.erb
create app/views/layouts/mailer.html.erb
create app/views/layouts/mailer.text.erb
```

Unit 01 | 로컬에서 Rails 시작하기

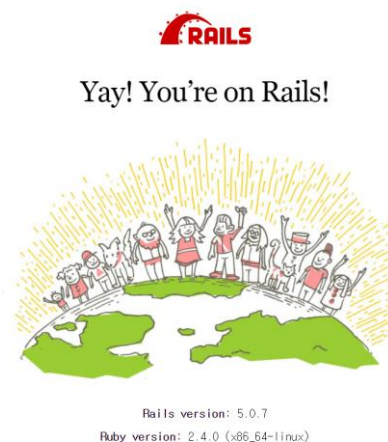
Rails 실행

생성한 프로젝트 폴더로 이동 후!

`rails s` (rails server)

localhost:3000 로 접속해보면

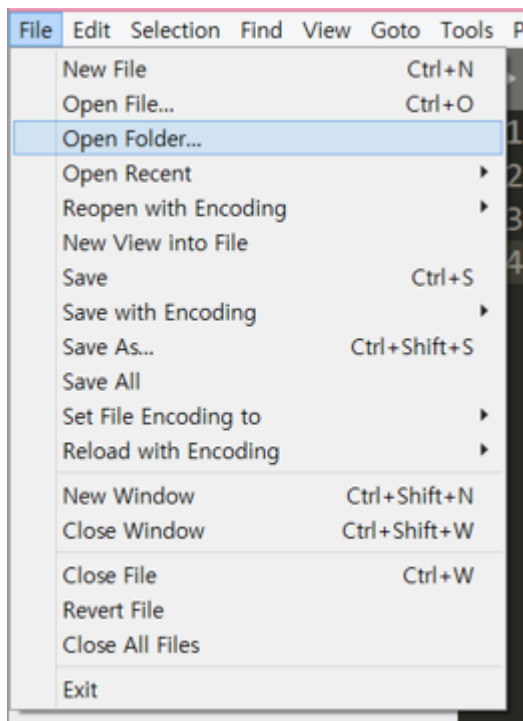
```
vagrant@ubuntu-xenial:~/vagrant$ cd w3_prac1
vagrant@ubuntu-xenial:~/vagrant/w3_prac1$ ls
app  config  db  Gemfile.lock  log  Rakefile  test  vendor
bin  config.ru  Gemfile  lib  public  README.md  tmp
vagrant@ubuntu-xenial:~/vagrant/w3_prac1$ rails s
=> Booting Puma
=> Rails 5.0.7 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
```



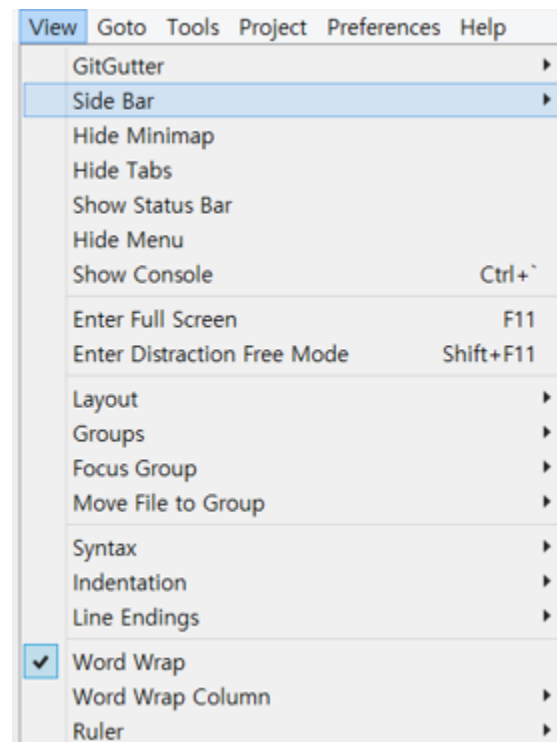
Unit 01 | 로컬에서 Rails 시작하기

SublimeText 에서 편리한 기능

로컬에서 vagrant 폴더 내의 project 폴더 선택

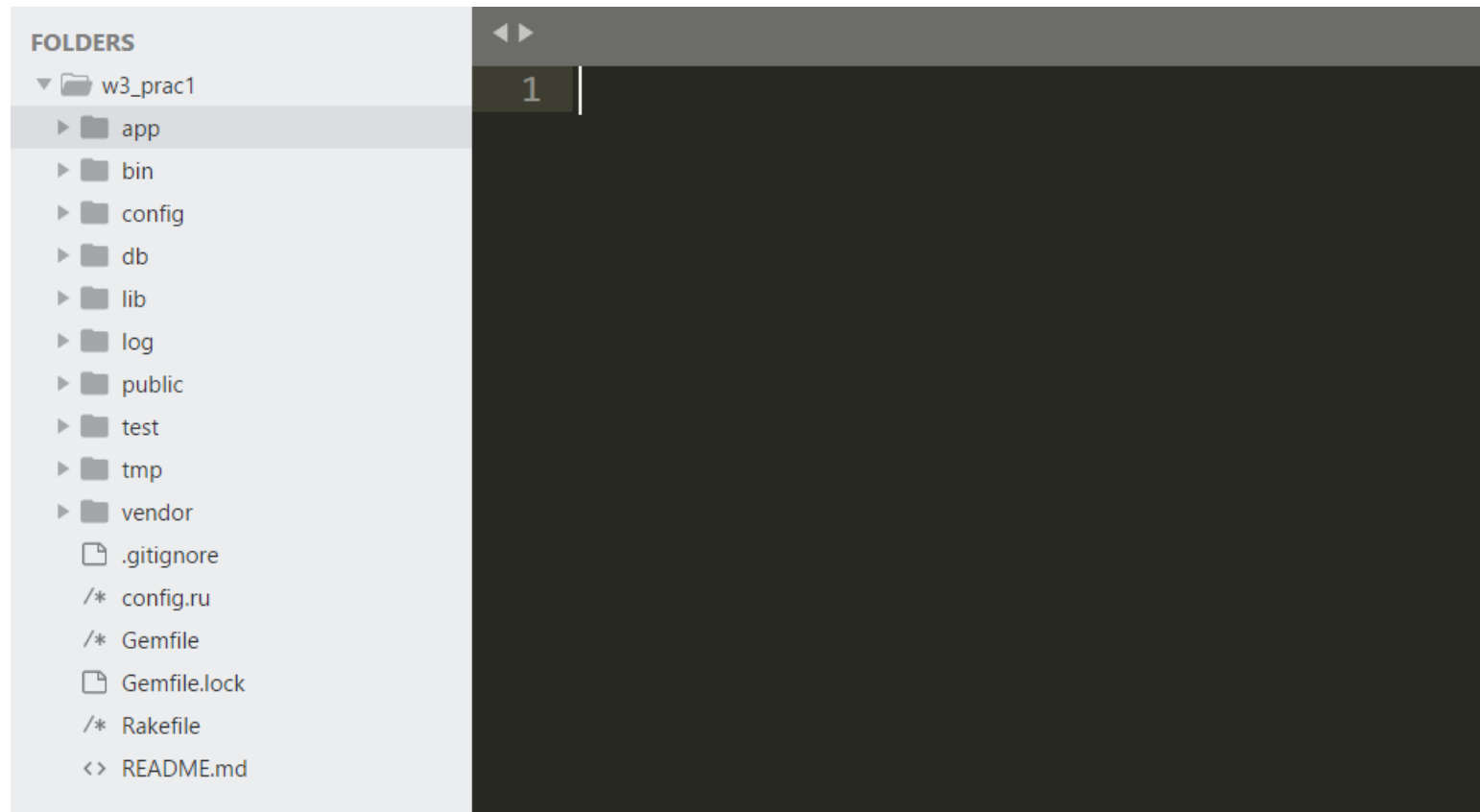


Side Bar 선택



Unit 01 | 로컬에서 Rails 시작하기

드디어 시작할 준비 완료..!



Unit 01 | 로컬에서 Rails 시작하기

FOLDERS

- ▼ w3_prac1
 - ▼ app
 - ▶ assets
 - ▶ channels
 - ▶ controllers
 - ▶ helpers
 - ▶ jobs
 - ▶ mailers
 - ▶ models
 - ▶ views
 - ▶ bin
 - ▼ config
 - ▶ environments
 - ▶ initializers
 - ▶ locales
 - /* application.rb
 - /* boot.rb
 - /* cable.yml
 - /* database.yml
 - /* environment.rb
 - /* puma.rb
 - /* routes.rb
 - /* secrets.yml
 - /* spring.rb

자주 사용하는 폴더 / 파일 설명

app folder (mvc)

- assets : html 파일에 필요한 소스들 (image, css 파일 등)
- controllers : controller들
- models : model들
- views : view 파일들

config folder

- routes.rb : 라우팅 설정

Unit 02 | VC 이해하기

Controller 생성

rails g controller [컨트롤러 이름]

: 기능별로 생성하는 것이 좋음 + 복수형태로 생성

rails g controller post (X)
rails g controller posts (O)

ex)

- posts controller : 게시글 기능 관리
- comments controller : 댓글 기능 관리

```
vagrant@ubuntu-xenial:/vagrant/w3_prac1$ rails g controller posts
Running via Spring preloader in process 2459
create    app/controllers/posts_controller.rb
invoke    erb
create    app/views/posts
invoke    test_unit
create    test/controllers/posts_controller_test.rb
invoke    helper
create    app/helpers/posts_helper.rb
invoke    test_unit
invoke    assets
invoke    coffee
create    app/assets/javascripts/posts.coffee
invoke    scss
create    app/assets/stylesheets/posts.scss
```

Unit 02 | VC 이해하기

Controller 의 역할

Action (상세기능) 을 관리함! : 함수형태로 존재

ex)

- Index 함수 : index 페이지에서 내용들을 어떻게 보여줄지 정함
- create 함수 : 게시글을 어떻게 생성할지 정함

Unit 02 | VC 이해하기

Controller 의 구성

Controller 는 **.rb 파일** => ruby 코드를 사용

```
posts_controller.rb • action1.html.erb • action3.html.erb × mailer.htm
class PostsController < ApplicationController

  def action1
    @name = "hyunkyung"
  end

  def action2
    @age = rand(20...25)
  end

  def action3
    @state = "hungry"
  end

end
```

@ 변수명 : **인스턴스 변수**

일반변수와 차이점 ?

: Controller 내 뿐만 아니라

프로젝트 내 ruby 코드를 사용하는 모든 파일에서 사용 가능

Unit 02 | VC 이해하기

View 생성

컨트롤러의 액션에 따라 보여줄 파일을 선택

컨트롤러 내의 함수 == view 파일 이름

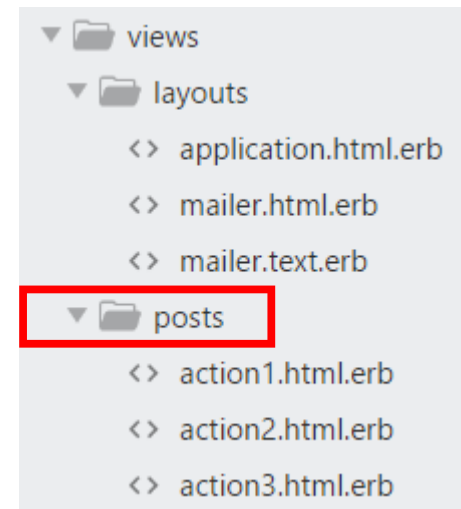
```
posts_controller.rb • action1.html.erb • action3.html.erb × mailer.htm
class PostsController < ApplicationController

  def action1
    @name = "hyunkyung"
  end

  def action2
    @age = rand(20...25)
  end

  def action3
    @state = "hungry"
  end

end
```



Unit 02 | VC 이해하기

View 의 구성

view는 .erb 파일 (embedded ruby) => ruby 코드를 사용가능

```
<% for i in 1..6 %>
  
<% end %>
```

화면에 보여야 하는 경우 <%= 코드 %>

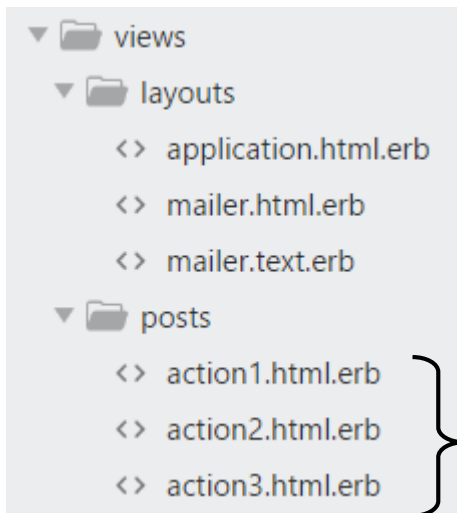
: 이미지나 텍스트, components 등에 주로 사용

화면에 보이지 않아도 되는 경우 <% 코드 %>

: 조건분 / 반복문 등에 주로 사용

Unit 02 | VC 이해하기

View는 어떻게 보일까?



이 erb 파일들은
<%= yield %> 자리에

```
application.html.erb x posts_controller.rb action1.html.erb action3.html.erb action2.html.erb x
<!DOCTYPE html>
<html>
  <head>
    <title>W3Prac1</title>
    <%= csrf_meta_tags %>

    <%= stylesheet_link_tag 'application', media: 'all', 'data-
    <%= javascript_include_tag 'application', 'data-turbolinks-tra
  </head>

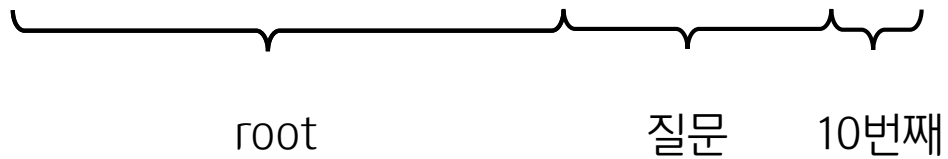
  <body>
    <%= yield %>
  </body>
</html>
```

Unit 02 | VC 이해하기

Route 설정

동적 웹의 디렉토리를 설정한다고 생각하면 됨

<https://uni.likelion.org/questions/10>



Unit 02 | VC 이해하기

Route 문법 다른 route 문법은 중간고사 끝나고 배울 예정입니다!

```
Rails.application.routes.draw do
  # For details on the DSL available here please visit https://guides.rubyonrails.org/routing.html
  root "posts#index"
end
```

root “[컨트롤러 이름]#[액션이름]”

Q & A

들어주셔서 감사합니다.