

0405

3주차 : CRUD#2

LIKE LION

EWHALION

5th 엄서영

3주차 : CRUD#2

CRUD

Create / Read / Update / Delete

간단한 게시판 기능



3주차 : CRUD#2

CRUD

Create / Read / **Update** / Delete

간단한 게시판 기능



CRUD#2[UD]

9

umseoyoung

Workspaces

Shared With Me

Repositories

YOUR INDIVIDUAL SUBSCRIPTIONS

Free Upgrade

Workspaces

Create a new workspace

crud2018
ruby workspace

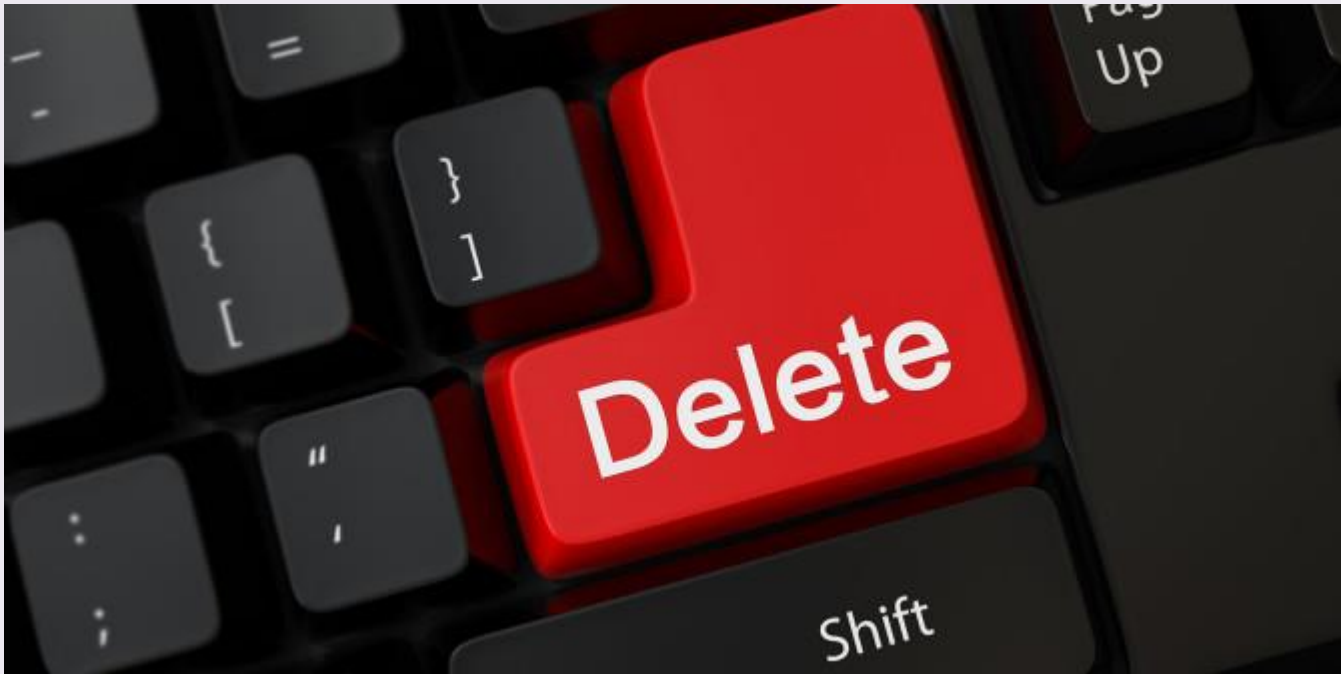
Add a description [here](#).

Updated a few seconds ago.

Clone Open

1 CPU 512MB RAM 2GB HDD

D! Delete!!!



1단계 : 삭제버튼 만들기

<index.html.erb>

```
<%@posts.each do |x|%>
  제 목 : <%=x.title%><br>
  내 용 : <%=x.content%><br>
  <a href='/home/destroy'>삭 제 </a><br>
  -----<br>
<%end%>
```

여기는 게시물 목록

새로운 글 쓰기

제목 :

내용 :

[삭제]

제목 :

내용 :

[삭제]

저기요 저기요
밑줄을 없애고 싶어요!!
방법을 알려주세요!



제목 :
내용 :
[삭제]

제목 :
내용 :
[삭제]



제목 :
내용 :
[삭제]

제목 :
내용 :
[삭제]

<index.html.erb>

```
<br><br>
<%@posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
<a href='/home/destroy' [redacted] >[삭제]</a><br>
-----<br>
<%end%>
```



<index.html.erb>

```
<br><br>
<%@posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
<a href='/home/destroy' style='text-decoration:none;'>[ 삭제 ]</a><br>
-----<br>
<%end%>
```





저기요 저기요
버튼 만든 다음에는
무엇을 해야 하나요?

Hint!

```
<br><br>
<%=posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
<a href='<b>/home/destroy</b>' style='text-dec
-----<br>
<%end%>
```

2단계 : route 작성하기

<config/routes.rb>

```
get 'home/destroy'  
#get 'home/destroy' => 'home#destory'
```


home/destroy 주소요청을 받으면
home 컨트롤러에 있는 destroy 액션을 수행하라

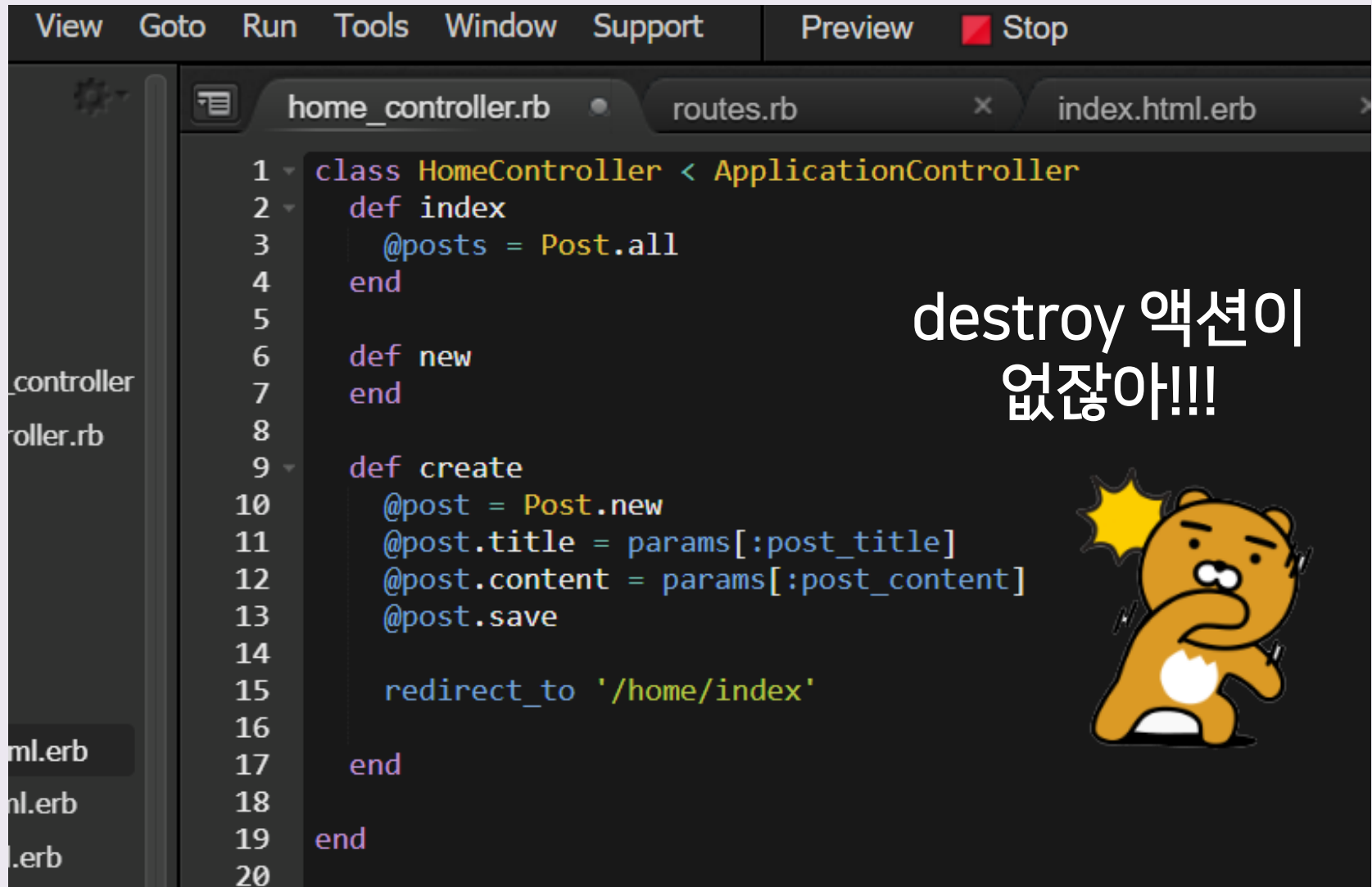
```
View  Goto  Run  Tools  Window  Support  Preview  Stop

home_controller.rb  routes.rb  index.html.erb

1  class HomeController < ApplicationController
2    def index
3      @posts = Post.all
4    end
5
6    def new
7    end
8
9    def create
10     @post = Post.new
11     @post.title = params[:post_title]
12     @post.content = params[:post_content]
13     @post.save
14
15     redirect_to '/home/index'
16
17   end
18
19 end
20
```


띠용쓰





```
1 class HomeController < ApplicationController
2   def index
3     @posts = Post.all
4   end
5
6   def new
7   end
8
9   def create
10    @post = Post.new
11    @post.title = params[:post_title]
12    @post.content = params[:post_content]
13    @post.save
14
15    redirect_to '/home/index'
16
17  end
18
19 end
20
```

destroy 액션이
없잖아!!!

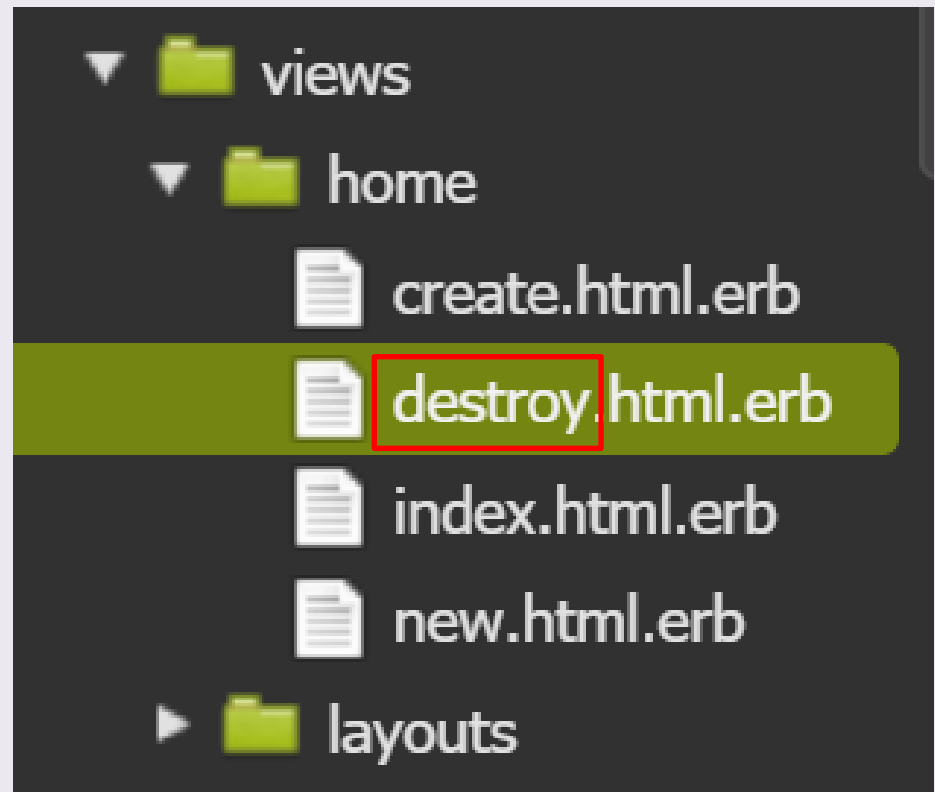


3단계 : destroy액션 / 뷰파일 만들기

<home controller>

```
def destroy  
end
```

수동생성!



4단계 : destroy액션에 코드 적기

<home controller>

```
def destroy
  Post.destroy_all
  redirect_to '/home/index'
end
```

삭제?

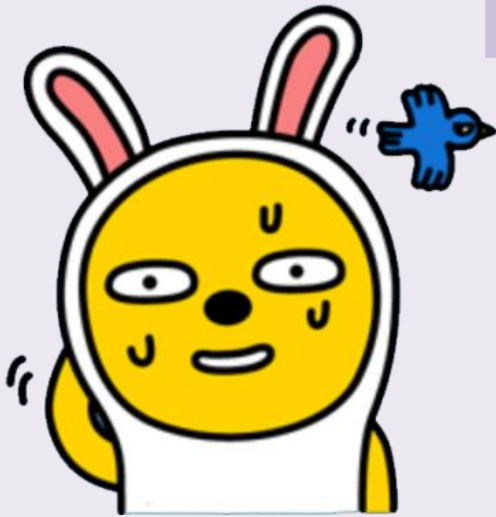
여기는 게시물 목록이 나올 페이지 입니다!

새로운 글 쓰기



띠용쓰





그 게시물
하나만 없애고 싶었는데...

다 없어짐!

```
tjdud0123:~/workspace $ rails c
Running via Spring preloader in process 4976
Loading development environment (Rails 4.2.5)
2.3.4 :001 > Post.all
Post Load (2.3ms) SELECT "posts".* FROM "posts"
=> #<ActiveRecord::Relation []>
2.3.4 :002 >
```

우리가 한 것 : 전체 게시물 삭제

우리가 원하는 것 : 특정 게시물 삭제

Id ÷ 주민등록번호

특정한 무언가를 나타내기 위한 수단

```
class CreatePosts < ActiveRecord::Migration
  def change
    create_table :posts do |t|

      t.string :title
      t.text :content

      t.timestamps null: false
    end
  end
end
```

※id는 자동적 생성

id	title	content

※id는 .save하면서 자동적 생성

```
2.4.1 :001 > Post.all
Post Load (0.9ms) SELECT "posts".* FROM "posts"
=> #<ActiveRecord::Relation [#<Post id: 9, title: "멋쟁이
사자처럼", content: "crud 강의", created_at: "2018-01-29 05
:15:28", updated_at: "2018-01-29 05:15:28">]>
```

id	title	contetnt
1		
2		
3		
4		
5		

Index 버튼/route 수정

<index.html.erb>

```

<h1>여기는 게시물 목록이 나올 페이지 입니다!</h1>
<a href='/home/new'>새로운 글 쓰기</a>

<br><br>
<%@posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
<a href='/home/destroy/<%=x.id%>' style='text-decoration:none;'>[삭제]</a><br>
-----<br>
<%end%>

```



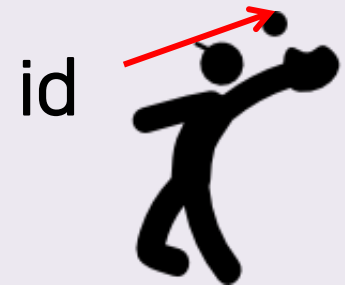
<config/routes.rb>

```

get 'home/destroy/:post_id' => 'home#destroy'

```

:id이름



EWha LiKELION

CRUD#2[UD]

```
get 'home/destroy/:post_id' => 'home#destroy'
```

:id이름 => :붙이는 것 중요! , 다시 넘겨야 하기 때문




```
> <br>  
oy/ :%=x.id%> ' s  
-----<br>
```

```
/:post_id'
```

```
Post.find(params[:post_id])
```

destroy 액션수정

```
def destroy
  Post.destroy_all
  redirect_to '/home/index'
end
```



```
def destroy
  post = Post.find(params[:post_id])
  post.destroy
  redirect_to '/home/index'
end
```

destroy 액션수정

```
def destroy
  post = Post.find(params[:post_id])
  post.destroy
  redirect_to '/home/index'
end
```

삭제해줘!

Post 모델에서
:post_id 값과 같은 값을 찾아줘!

CRUD#2[UD]

```
get 'home/index'
```

```
class HomeController < ApplicationController
  def index
    @posts = Post.all
  end
end
```

지우고 난 나머지 데이터들을 다시
`posts` 변수에 저장

```
<h1>여기는 게시물 목록이 나올 페이지 입니다!</h1>
<a href='/home/new'>새로운 글 쓰기</a>

<br><br>
<%= @posts.each do |x| %>
제목 : <%= x.title %> <br>
내용 : <%= x.content %> <br>
-----<br>
<%end%>
```

재출력

여기는 게시물 목록

새로운 글 쓰기

제목 : 안녕
내용 : 와우웅와
[삭제]

제목 : 끼요우우웅
내용 : 후헤헤헤헥
[삭제]

여기는 게시물

새로운 글 쓰기

제목 : 안녕
내용 : 와우웅와
[삭제]



성공!

D! Delete 4단계

1단계 : 삭제버튼 만들기

2단계 : route 작성하기

3단계 : 액션 / 뷰파일 만들기

4단계 : 액션에 코드 적기

U! Update!!!



edit : 덮어쓰기

새 글을 쓰는 것(new) : 빈 테이블을 만들고 내용을 채우는 과정

글을 수정하는 것(edit) : 채워진 테이블의 내용을 바꾸는 과정

수정페이지 레이아웃 = 새 글 작성 레이아웃

기존 글 수정 페이지입니다

제목 :

내용 :

edit.html.erb

=

새로운 글 작성 페이지입니다

제목 :

내용 :

new.html.erb

단, **이전 내용**을 불러와야 함

여기는 게시물 목록

[새로운 글 쓰기](#)

제목 : 안녕
내용 : 와우웅와
[\[삭제\]](#)
[\[수정\]](#)

수정하면서 id는 두 번 필요

여기는 게시물 목록

[새로운 글 쓰기](#)

제목 : 안녕
내용 : 와우웅와

[삭제]
[수정]

기존 글 수정 페이지입니다

제목 :

내용 :

제출

제목 : 끼요우우웅
내용 : 후헤헤헤헹
[삭제]

수정하는 양식에 **이전에 썼던 내용**을 불러올 때

이전 글을 수정한 뒤 새로운 내용으로 **업데이트** 할 때

1단계 : 수정버튼 만들기

<index.html.erb>

```
<br><br>
<%@posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
<a href='/home/destroy/<%=x.id%>' style='text-decoration:none;'>[삭제]</a><br>
<a href='/home/edit/<%=x.id%>' style='text-decoration:none;'>[수정]</a><br>
-----<br>
<%end%>
```

여기는 게시물 목록

새로운 글 쓰기

제목 : 안녕

내용 : 와우웅와

[삭제]

[수정]

2단계 : edit route 작성하기

<config/routes.rb>

destroy와 비슷!

```
get 'home/edit/:post_id' => 'home#edit'
```

home/edit 주소요청을 받으면

home 컨트롤러에 있는 edit 액션을 수행하라



저기요 저기요
라우팅한 다음에는
무엇을 해야 하나요?

Hint!

```
get 'home/edit/:post_id' => 'home#edit'
```

저기요 저기요
라우팅한 다음에는
무엇을 해야 하나요?



Hint!

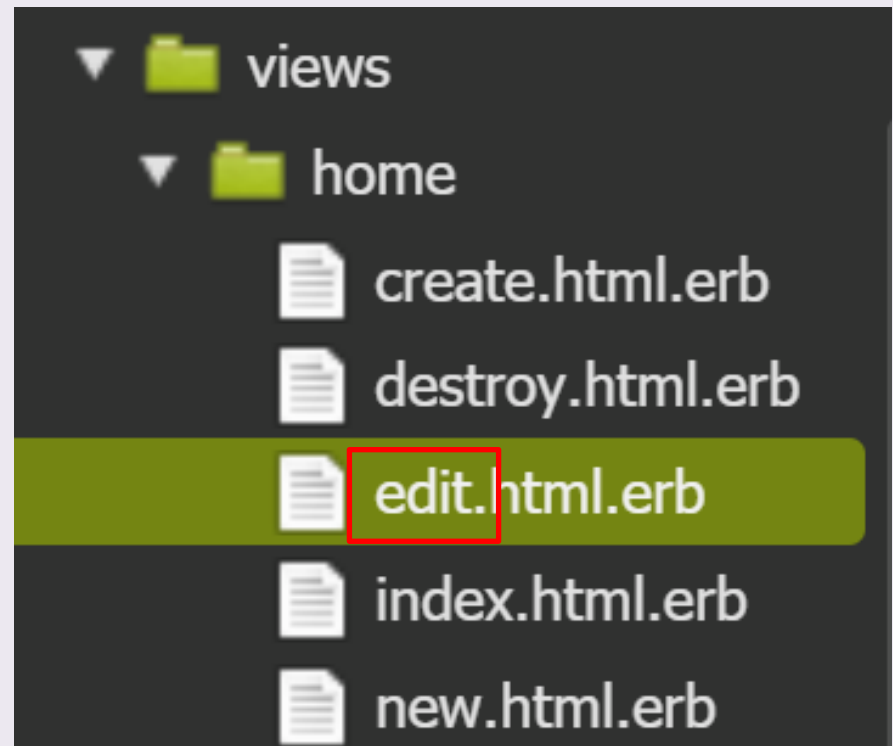
```
get 'home/edit/:post_id' => 'home#edit'
```



3단계 : edit 액션 / 뷰파일 만들기

<home controller>

```
def edit  
end
```



4단계 : edit 액션에 코드 적기

```
def edit  
  post =   
end
```



뭘 적지?
수정 버튼 누른 **그 게시물**
을 받아와야 할 텐데...

4단계 : edit 액션에 코드 적기

<home controller>

```
def edit
  post = Post.find(params[:post_id])
end
```



```
def edit
  @post = Post.find(params[:post_id])
end
```

차이점...?

CRUD#2[UD]

```
<h1>여기는 게시물 목록이 나올 페이지입니다</h1>
<a href='/home/new'>새로운 글 쓰기</a>

<br><br>
<%= @posts.each do |x| %>
제목 : <%= x.title %> <br>
내용 : <%= x.content %> <br>
-----<br>
<%end%>
```

```
class HomeController < ApplicationController
  def index
    @posts = Post.all
  end
end
```

<home_controller.rb>

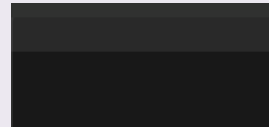
Views/home/<Index.html.erb>

뷰파일에서 사용하고 싶으면
@를 붙여서 **인스턴스 변수**를 꼭 사용

즉, 뷰파일에서 사용하고 싶은 변수라면
@도장광 찍어서 **너는 뷰파일에 쓸거야 변수야!**
라고 말해주면 됨

CRUD#2[UD]

```
def destroy
  post = Post.find(params[:post_id])
  post.destroy
  redirect_to '/home/index'
end
```



<destroy.html.erb>

destroy view파일에서 사용하지 않음.

기존 글 수정 폼

제목 :

내용 :

```
def edit
  @post = Post.find(params[:post_id])
end
```

edit view파일에서 사용 함.

```
<%= @post.content %>
```

<edit.html.erb>

5단계 : edit(글 수정 페이지) view파일 작성하기

<edit.html.erb>

```
<h1>기존 글 수정 페이지입니다</h1>
<form action='/home/update/<%=@post.id%>' method='post'>
  <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
  제목 : <input type='text' name='post_title' value='<%=@post.title%>'><br><br>
  내용 : <textarea name='post_content' ><%=@post.content%></textarea><br><br>
  <input type='submit' value='제출'>
</form>
```

새로운 글 쓰기

제목 : 안녕
 내용 : 와우웅와
 [삭제]
 [수정]

기존 글 수정 페이지입니다

제목 : 안녕
 내용 : 와우웅와
 제출

6단계 : update route 작성하기

<config/routes.rb>

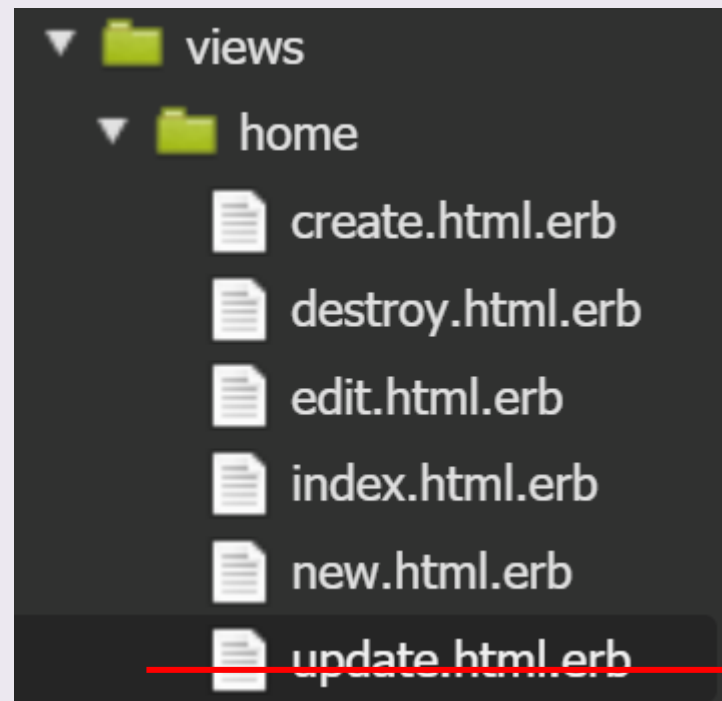
```
post 'home/update/:post_id' => 'home#update'
```

7단계 : update 액션 /뷰파일 만들기(안 쓰이므로 안 해도 됨)

<home controller>

```
def update  
end
```

```
redirect_to '/home/index'
```



8단계 : update 액션에 코드 적기

<config/routes.rb>

<home controller>

```
post 'home/update/:post_id' => 'home#update'
```

```
def update
  post = Post.find(params[:post_id])
  post.title = params[:post_title]
  post.content = params[:post_content]
  post.save

  redirect_to '/home/index'
end
```

찾아와서 저장

```
pe='text' name='post_title'
name='post_content' ><%=@p
mit' value='제출'>
```

<edit.html.erb>

게시물의 내용을 바꿔치기 후
저장!

U! Update 7단계

1단계 : 수정버튼 만들기

2단계 : edit route 작성하기

3단계 : edit 액션 / 뷰파일 만들기

4단계 : edit 액션에 코드 적기

5단계 : edit(글 수정 페이지) view파일 작성하기

6단계 : update route 작성하기

7단계 : update 액션 /뷰파일 만들기(안 쓰이므로 안 해도 됨)

8단계 : update 액션에 코드 적기

CRUD

Create / Read / Update / Delete

간단한 게시판 기능

다음시간 까지
CRUD 꼭 복습~

CRUD#2[UD]

넘모 넘모
수고하셨습니다~!

