



**LIKE LION**

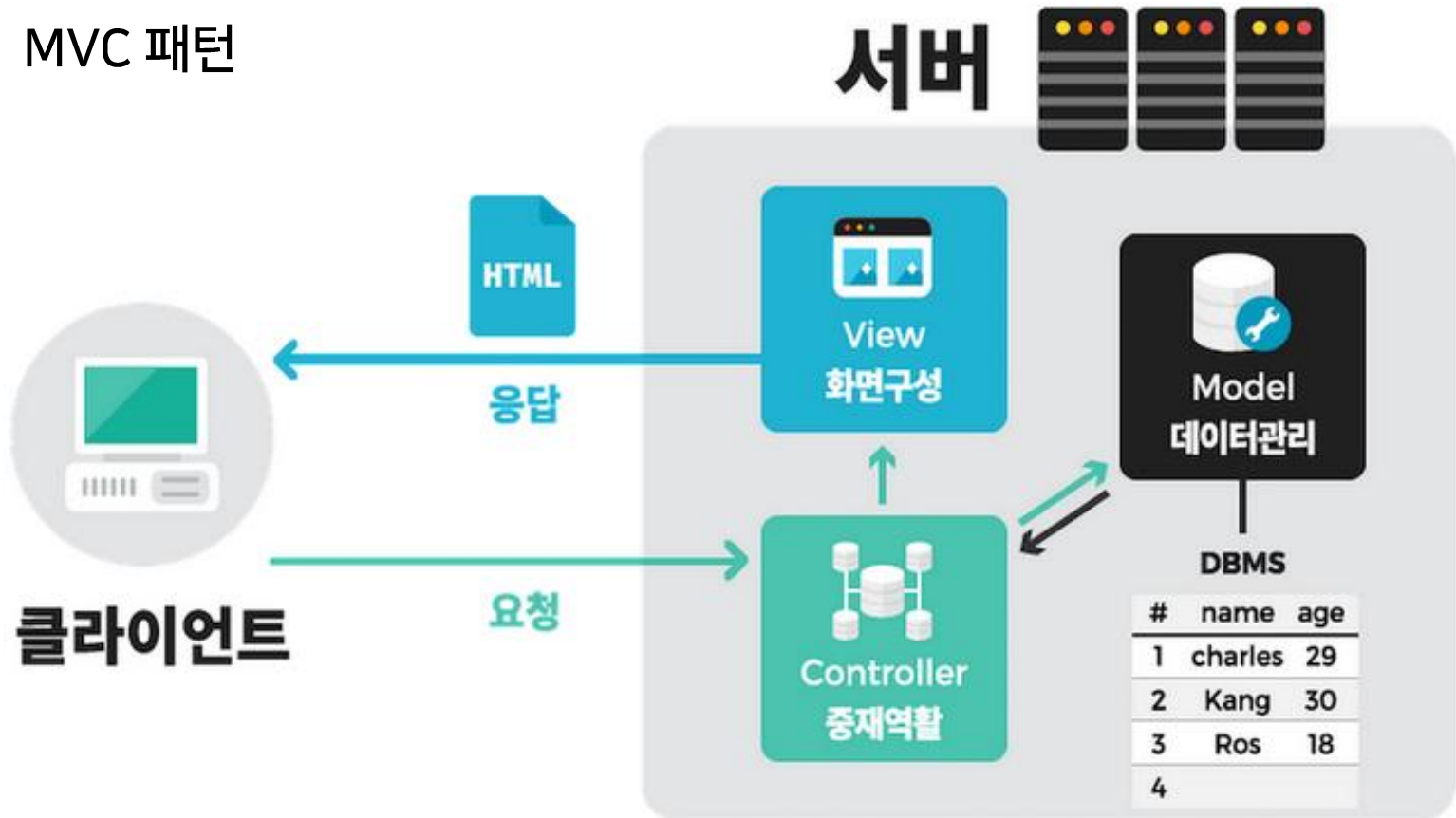
**EWHALION**

2주차 : MVC & Route

이화여대 멋사 김준선

# MVC & Route

## MVC 패턴



# MVC & Route



'이화여대'  
궁금해!!!

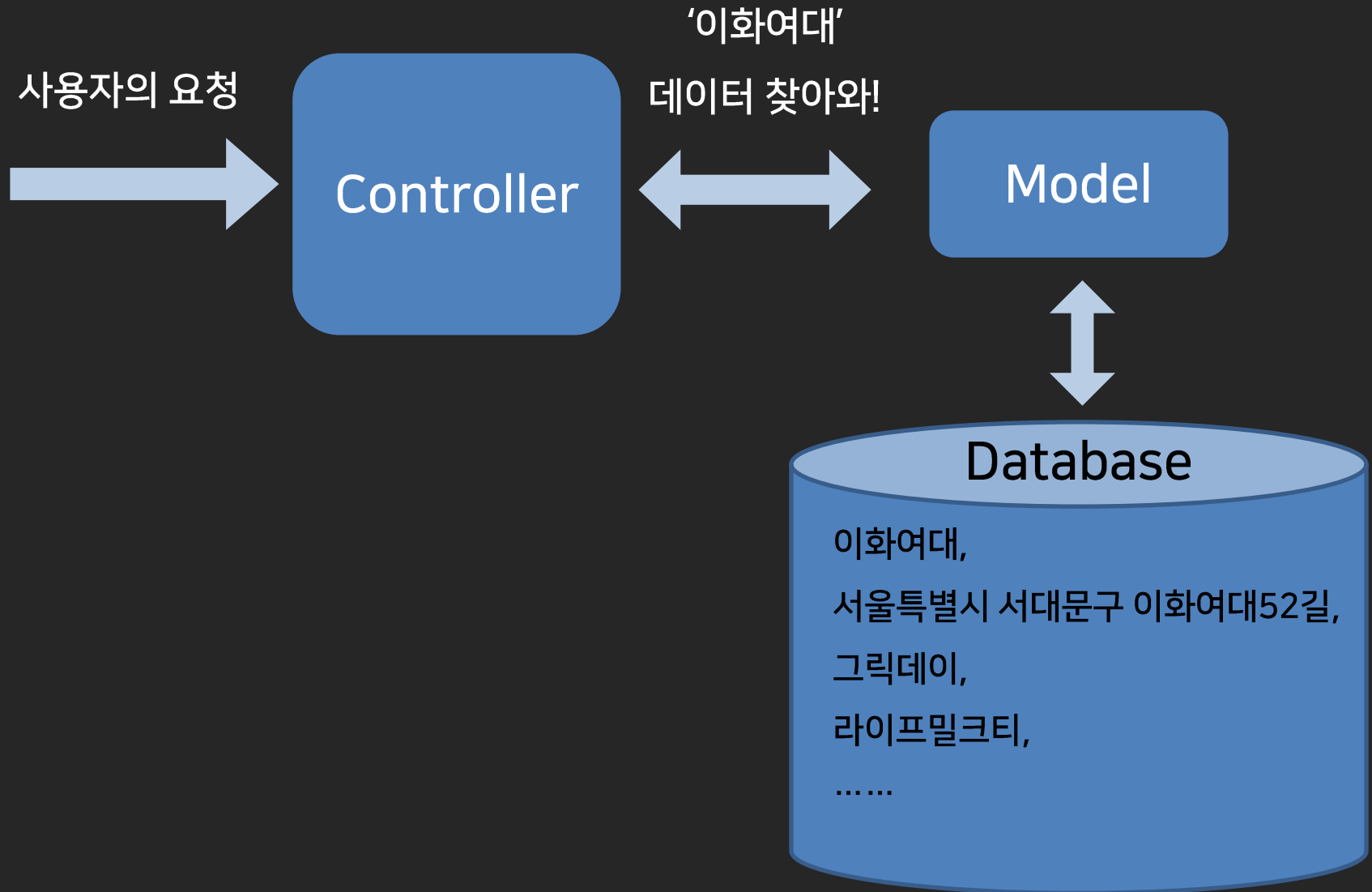
**NAVER**

이화여대

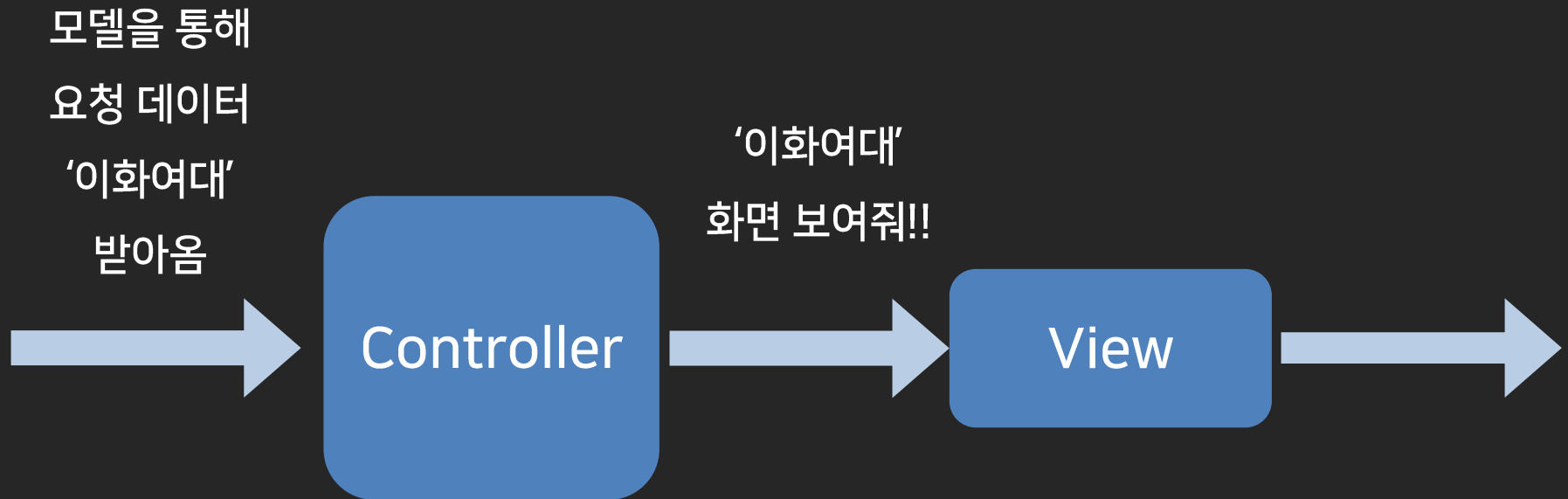


사용자가 요청을 보낸다!

# MVC & Route



# MVC & Route

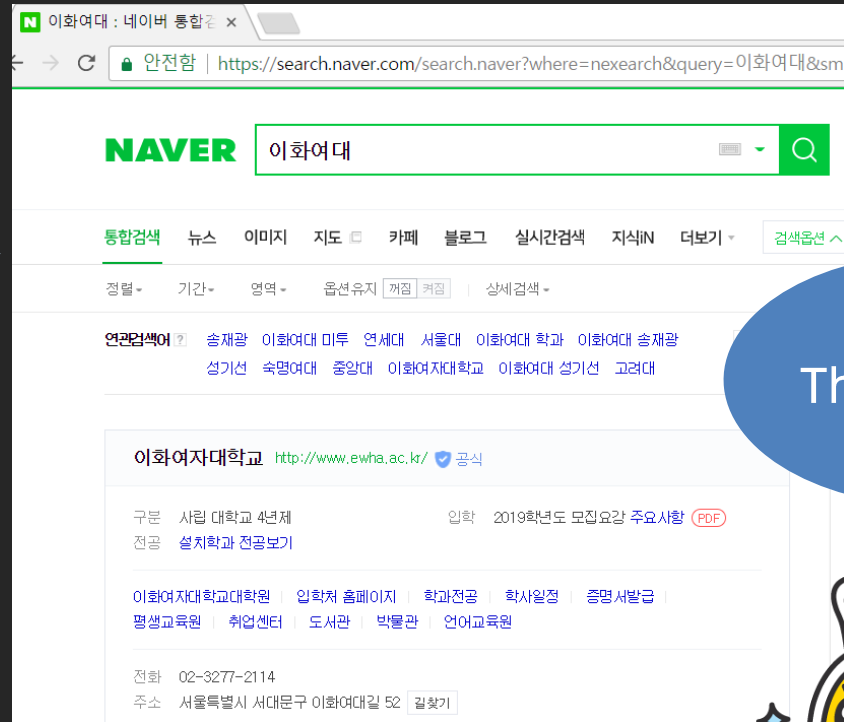


# MVC & Route

## <HTML 파일>

View

응답!



Thank you !!



## 정리하자 MVC

**M (Model)** : 데이터베이스를 다루는 부분

데이터베이스 테이블과 관련된 규칙 기능 등을 정의  
데이터 생성, 조회, 수정

**V (View)** : 사용자에게 직접 보여지는 부분

html/css/javascript 등 Front-end 구성요소

**C (Controller)** : Model 과 View 사이에서 둘의 상호작용을 중재

요청에 맞는 모델로 부터 데이터조회, 뷰에 반영

## MVC

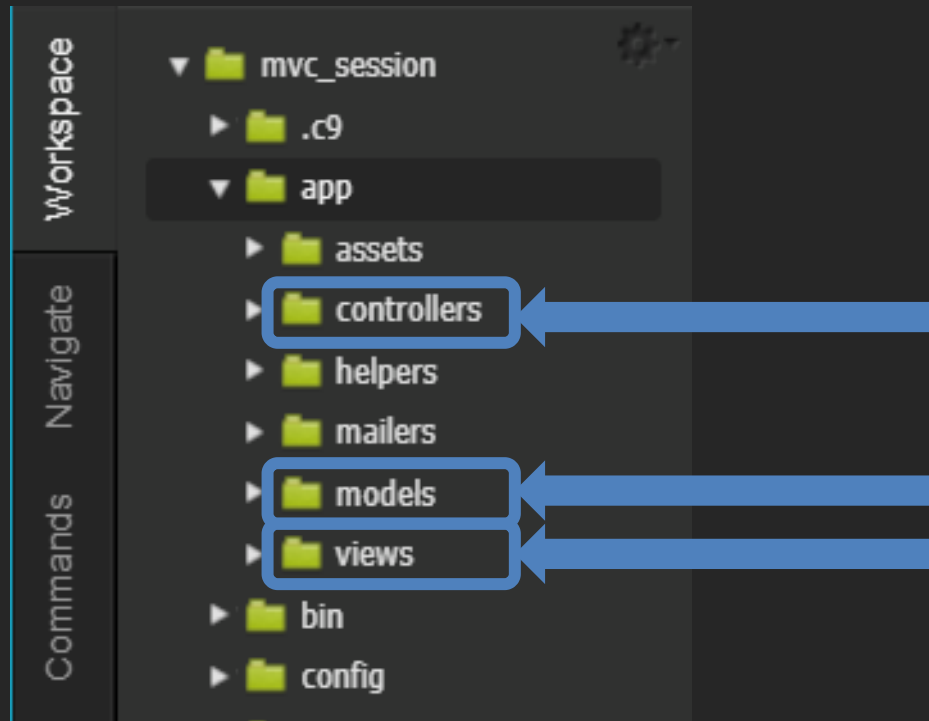
별거 아니었어!

그럼 직접 해봅시다!





## C9 에서 Ruby on Rails MVC



우리 이제 친해지기로 해요...😊

A terminal window with a dark blue background. The title bar shows 'bash - "zzunvely-mv' and 'Immediate'. The prompt is 'zzunvely:~/workspace \$' followed by a cursor.

명령어 창! 모든 명령어는 이곳에!

레일즈 개발의 시작은 **컨트롤러** 만들기

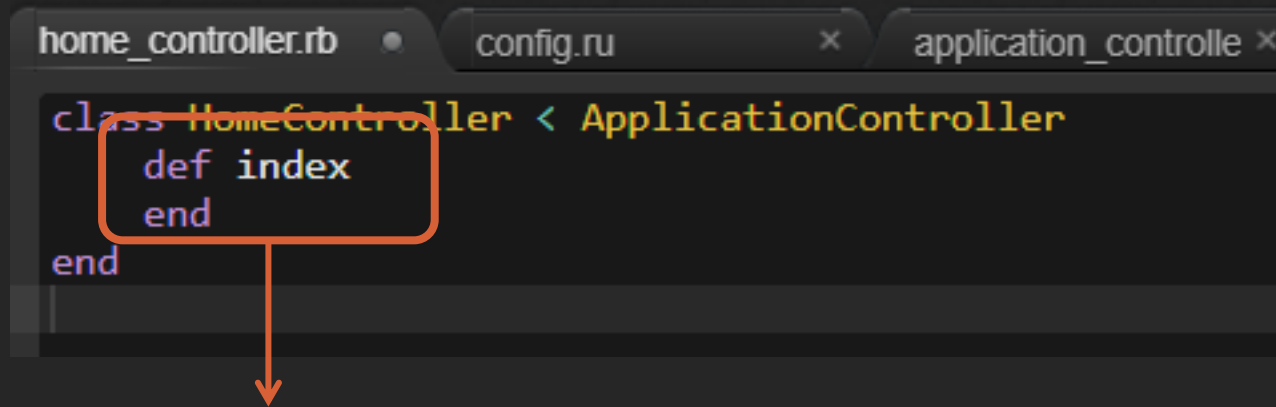
rails g controller [컨트롤러이름]

=> g는 generate 의 의미!

```
:~/workspace $ rails g controller home
```

# MVC & Route

S  
th  
6



컨트롤러는 **액션**들의 집합

Action : 사용자로부터 받은 **요청**

직접 **처리**하는 친구!

# MVC & Route

S  
th  
6

```
class HomeController < ApplicationController
  def index
    # 모든 게시물 보여주기
  end
  def create
    # 게시물 생성해달라는 요청을 처리할 액션
  end
  def show
    # 1개 게시물 보여줘!
  end
  def destroy
    # 게시물 삭제해줘!
  end
  def edit
    # 게시물 수정 요청!
  end
end
```

특정 요청 => 특정 처리    By. 특정 **액션!**

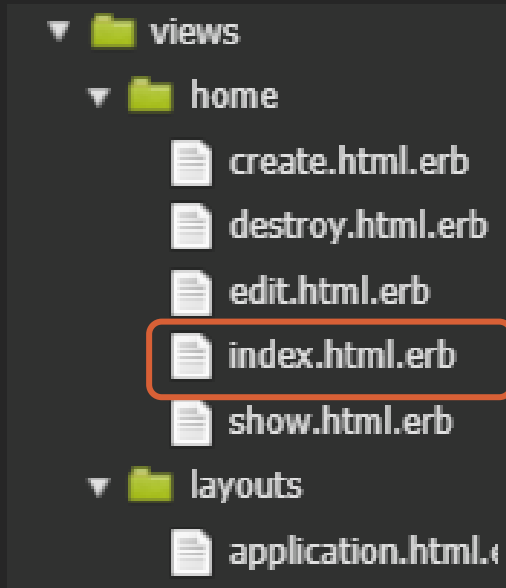
## 특정 action 에 대한 View 파일 만들기

```
class HomeController < ApplicationController
  def index
    # 모든 게시물 보여주기
  end
  def create
    # 게시물 생성해달라는 요청을 처리할 액션
  end
  def show
    # 1개 게시물 보여줘!
  end
  def destroy
    # 게시물 삭제해줘!
  end
  def edit
    # 게시물 수정 요청!
  end
end
```

```
views
├── home
│   ├── create.html.erb
│   ├── destroy.html.erb
│   ├── edit.html.erb
│   └── index.html.erb
└── layouts
    └── application.html.erb
```

# MVC & Route

S  
th  
6



.html.erb 파일?

erb = Embedded Ruby

HTML 파일 안에서 Ruby 코드 가능!

그러면 이제...



# Route(라우트)

config/route.rb

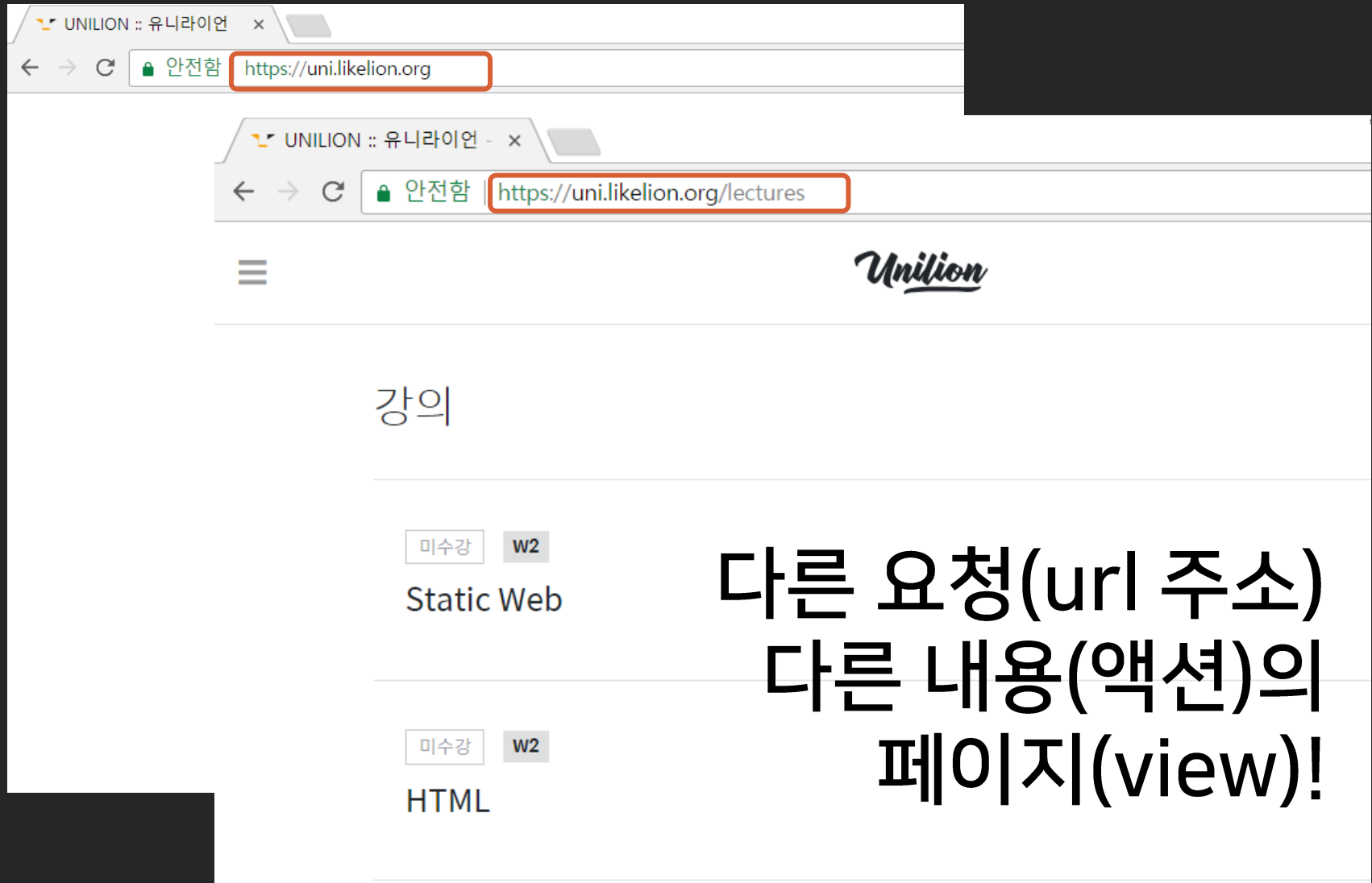
: URL 요청을

특정 컨트롤러의 특정 액션(뷰파일)로

매칭/ 연결

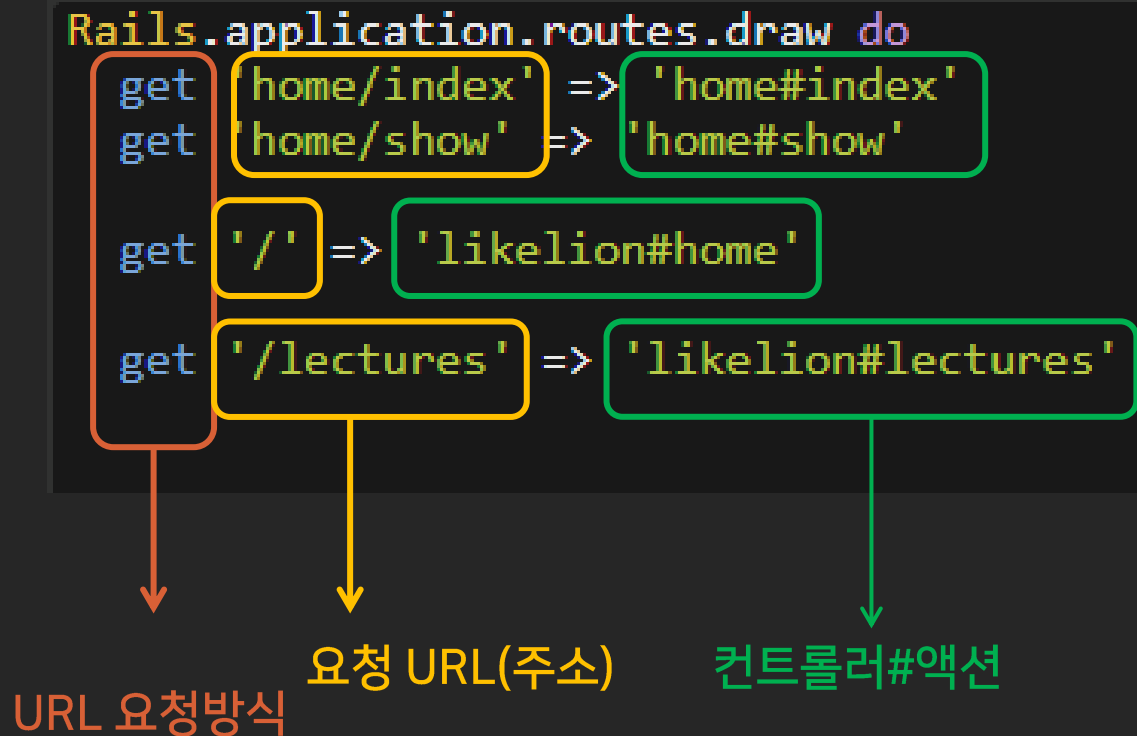
# MVC & Route

S  
th  
6



# MVC & Route

S  
th  
6



즉, uni.likelion.org => 원래페이지  
uni.likelion.org/lectures

## URL 요청방식

get

URL 뒤에 데이터가 추가되어 전달  
입력 값이 간단하거나,  
공개적으로 url을 넘겨도 무방한 경우

가져오기, 조회

간단한 데이터 처리  
빠름~

post

URL과 별도로  
http에서 데이터 전송

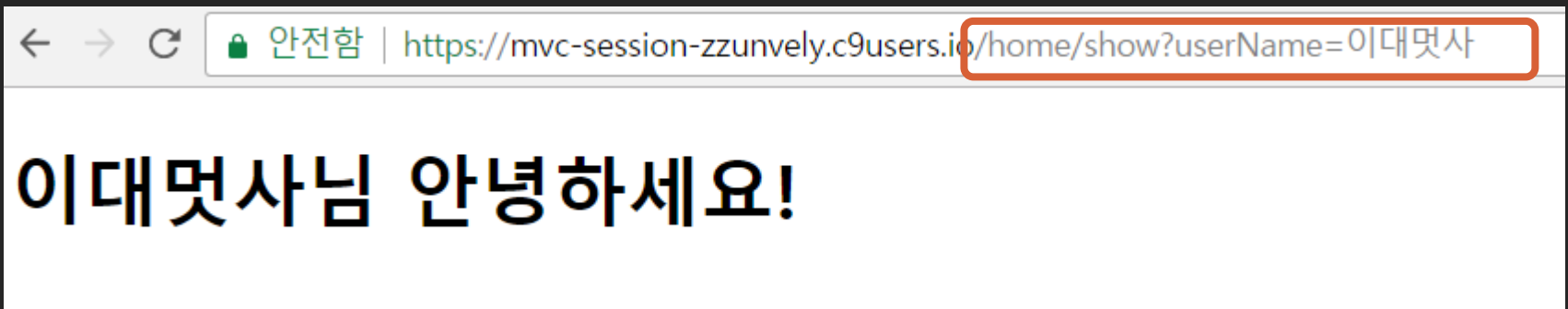
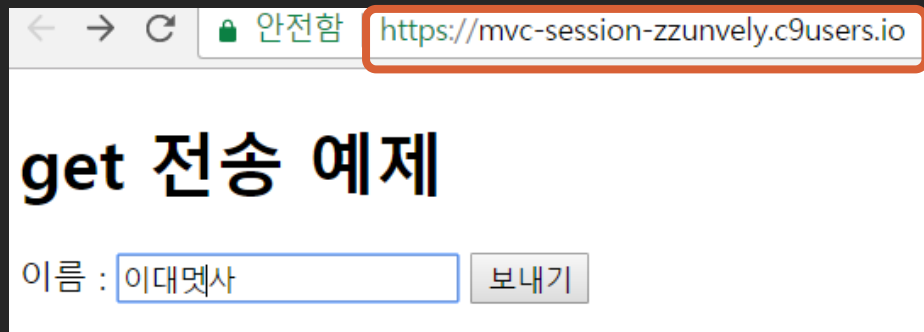
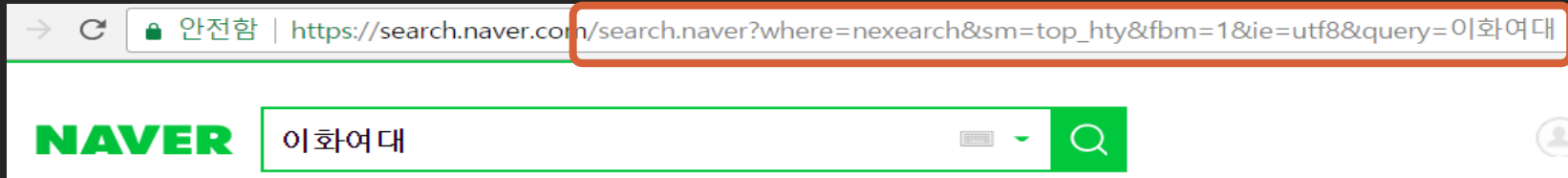
수행하기, 제출

Get보다 많은 데이터 전송  
(파일이나 게시물)  
최소한의 보안유지가능

# MVC & Route

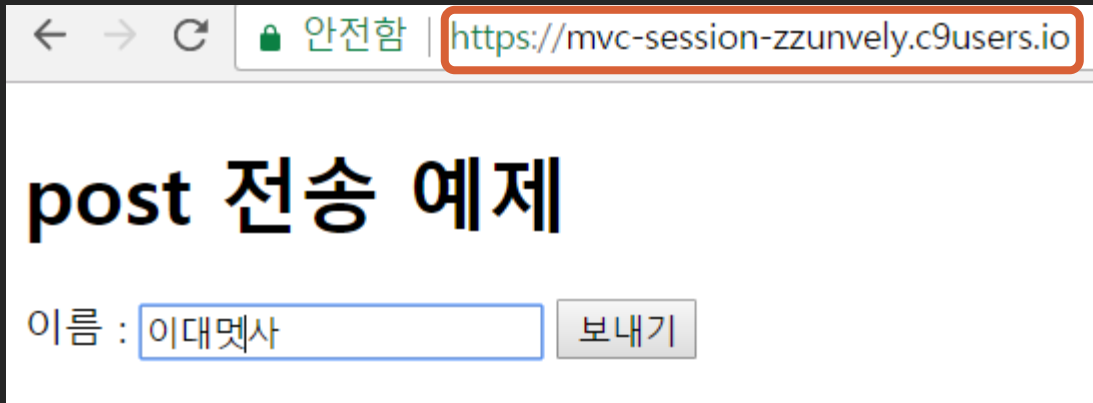
get

S  
th  
6



# MVC & Route

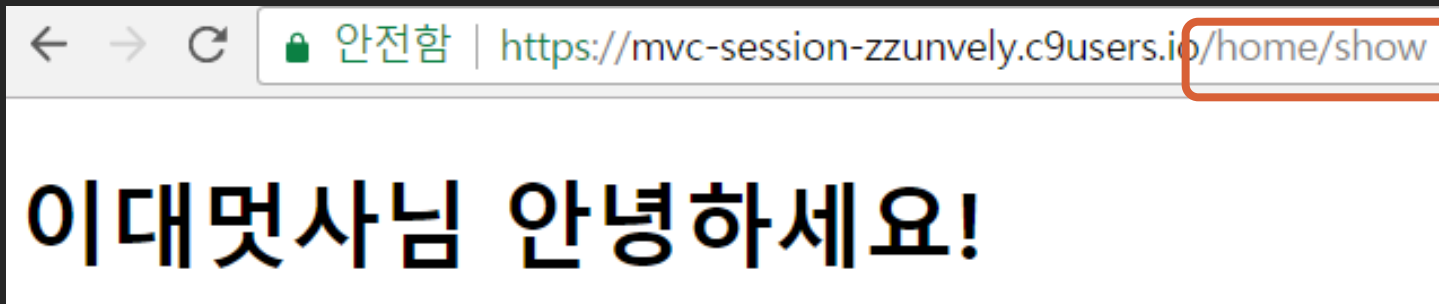
post



← → ↻ | 안전함 | https://mvc-session-zzunvely.c9users.io

## post 전송 예제

이름 :



← → ↻ | 안전함 | https://mvc-session-zzunvely.c9users.io/home/show

## 이대멋사님 안녕하세요!

# MVC & Route

root

```
get '/' => 'home#index'  
root 'home#index'
```

똑같다!

root '[controller]#[action]'

처음 접속했을 때

해당 [controller]#[action] 으로 요청 보내는 것!

(주의!! 반드시 1개)

root 도 get 방식!

## 그럼 여기서 Quiz

'ewha.likelion.org' 페이지의  
Config/route.rb 파일의 일부분이 아래와 같다면!

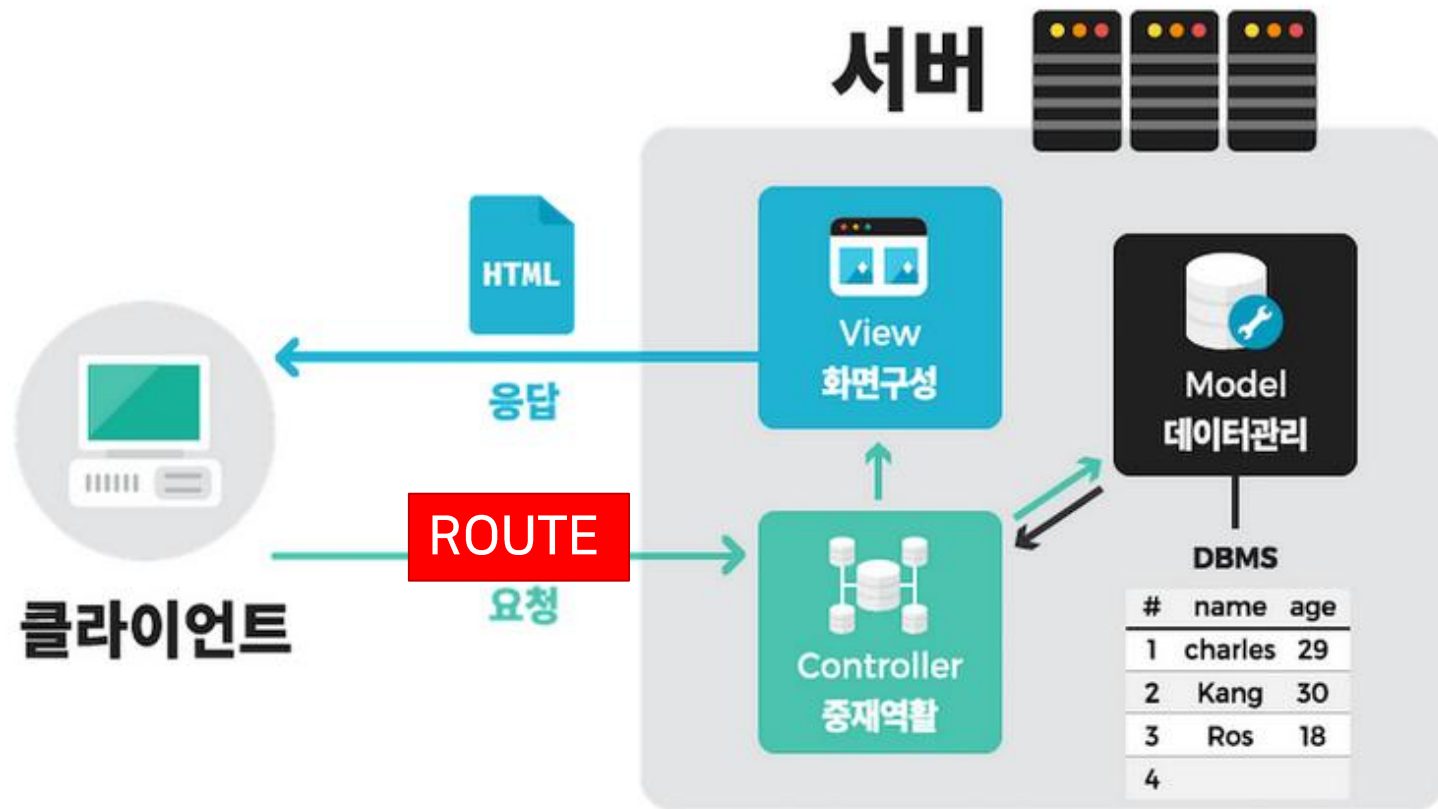
```
get 'likelion' => 'home#staff'
```

Staff 뷰페이지를 보고 싶다면  
어느 주소로 접속해야 할까요?



# MVC & Route

S  
th  
6



기억하자!

컨트롤러 만들기

액션 만들기

뷰(액션에 해당하는) 파일 만들기

라우트 연결하기

# MVC & Route

깨알 꿀팁!

S  
th  
6

rails g controller [이름] [action이름] [action이름] ...

컨트롤러 생성, 그 안에 액션정의, 뷰파일 생성, routing 까지  
한꺼번에 다해줌!

```
zzunvely:~/workspace $ rails g controller ewha home index
```

```
class EwhaController < ApplicationController
  def home
  end

  def index
  end
end
```

```
▼ views
  ▼ ewha
    home.html.erb
    index.html.erb
```

```
Rails.application.routes.draw do
  get 'ewha/home'
  get 'ewha/index'
```

# MVC & Route

S  
th  
6



뷰(html파일)에서 사용자 입력 받아오기!

## 사용자 입력 by. form 태그

- 입력된 데이터를 한꺼번에 서버에 전송하기 위해 사용
- form 태그 속성

method : 어떤 방식으로 전달?

ex) method = get, method = post

action : 어디로 전달?

ex) action = '/home/index'

## <뷰파일>

```
<form action="/result" method="post" >  
  <input type="text" name="title" align="middle">  
  <input type="text" name="content" align="middle">  
  <input type="submit" value="전송" >  
</form>
```

입력 받은 form 안의 정보를  
post 방식으로  
'/result'(url)으로 보내줘!

```
<form action="/result" method="post" >
```

```
<input type="text" name="title" align="middle">  
<input type="text" name="content" align="middle">  
<input type="submit" value="전송" >
```

```
</form>
```

<input 태그>


주로 form 안에 들어가서  
실질적으로 입력된 정보를 받아주는 친구

\* Input 타입은 저절로 닫힘!

# MVC & Route

S  
th  
6

```
<form action="/result" method="post" >  
  <input type="text" name="title" align="middle">  
  <input type="text" name="content" align="middle">  
  <input type="submit" value="전송" >  
</form>
```



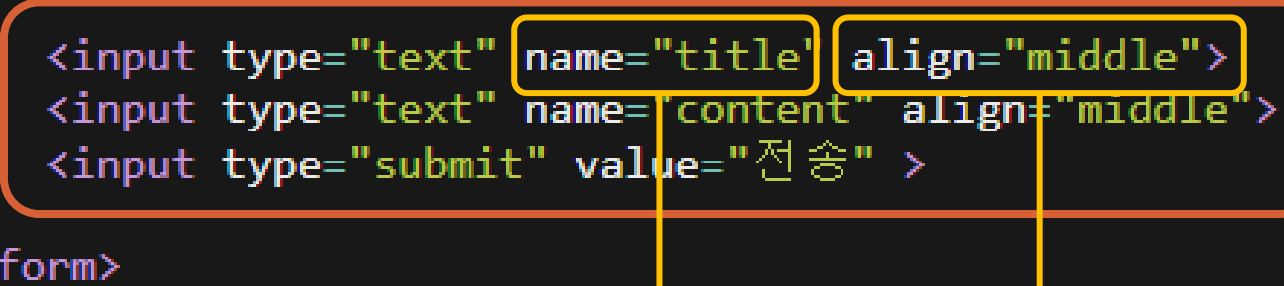
Input tag의 종류 : 텍스트를 입력 받을거야!



# MVC & Route

S  
th  
6

```
<form action="/result" method="post" >  
  <input type="text" name="title" align="middle">  
  <input type="text" name="content" align="middle">  
  <input type="submit" value="전송" >  
</form>
```




입력 받은 정보의 이름!     가운데 정렬~

# MVC & Route

S  
th  
6

```
<form action="/result" method="post" >  
  <input type="text" name="title" align="middle">  
  <input type="text" name="content" align="middle">  
  <input type="submit" value="전송" >  
</form>
```



입력한 값 제출!

서버에 form 값 전송

Value는 전송 버튼 위에 쓰여질 값!

## Input 태그 type

1. text : 텍스트를 입력할 수 있는 text field  
(많은 양의 text를 입력할 때는 text-area 태그 사용 ex) 게시글의 내용)
2. submit : 전송버튼 . 전송버튼은 form 요소에서 정해진 대로 폼을 서버에 전송
3. reset : 초기화 버튼. 버튼을 누르면 입력된 값이 초기화
4. button : 버튼이 만들어 짐
5. password : 입력내용이 \* 로 대체 됨
6. file : 파일 업로드 컨트롤이 만들어 짐
7. checkbox : 체크 박스가 만들어짐
8. email : 이메일 주소 규정에 맞추어 입력해야 함

더 많은 type들은 링크 참고!

[https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp)

# MVC & Route

S  
th  
6



컨트롤러의 액션 => 사용자가 보낸 정보 받아오기!

## params

:POST방식으로 사용자가 보낸 정보에  
접근할 수 있도록 해주는 해시

params[:form의name태그이름]

params["formd의 name태그이름"]

# MVC & Route

S  
th  
6

```
class HomeController < ApplicationController
  def index
  end

  def result
    @title1=params[:title]
    @content1=params[:content]
  end
end
```

사용자가 보낸 정보 받아와!

# MVC & Route

S  
th  
6

```
class HomeController < ApplicationController
  def index
  end

  def result
    @title1=params[:title]
    @content1=params[:content]
  end
end
```

정보 받아서 @title1  
@content1 에 담아줘!

인스턴스변수 @

```
@title1=params[:title]  
@content1=params[:content]
```

:

컨트롤러에서 연산된 결과를

해당 view(해당 액션)에서 보여주기 위해서!



# MVC & Route

[Home controller 파일]

```
class HomeController < ApplicationController
  def index
  end

  def result
    @title1=params[:title]
    @content1=params[:content]
  end
end
```

[result.html.erb 파일]

```
<h2>제목 : <%= @title1%></h2>
<h2>내용 : <%= @content1%></h2>
```

## [간단한 실습]

나와 내 옆자리 사람을 입력받아  
출력해주는기!

### 오늘의 내 짝꿍!

나는 :

오늘 내 옆에는 :

### Home#show

나는 : 아기사자

내 짝꿍: 라이온킹

## [간단한 실습]

1. home controller와 index, show 2개의 액션 만들기  
(컨트롤러, 액션 만들면 뭐하랬지?)
2. Index 뷰페이지에 form, input 형태로 정보 입력 받기
3. form tag로 인한 post값 route해주기
4. home 컨트롤러 show 액션에 정보 받아오기
5. show 뷰페이지에 출력!

VC는 만들었고...  
그럼 이제 M만 남았네!

나머지 실습은  
추가과제!  
랜덤게임~ 랜덤게임~  
아기사자가 좋아하는  
랜덤게임~



수고하셨습니다요 😊

