

게시판 글쓰기

—



글쓰기 화면

글쓰기 화면

제목:

내가 이 수업을 듣는 이유...

사실상 우리가 이걸 듣는 이유는 뭐 잘 모르겠다는 거..
궁시렁 궁시렁..

올리기

프로젝트 생성

—

템플릿 프로젝트 생성

1. Eclipse에서 Git Repository Exploring Perspective로 이동한다.
2. <https://github.com/javajigi/next-board.git> 를 추가한다.
3. next-board 프로젝트를 import 한다.
4. pom.xml 오른쪽 클릭 > Run As > Maven build ... > Goals에 eclipse:eclipse
입력 후 Run 실행
5. next-board 프로젝트에서 F5 실행(또는 오른쪽 클릭)해 Refresh

Compiler 버전과 인코딩 지정

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.3.2</version>
  <configuration>
    <source>1.6</source>
    <target>1.6</target>
    <encoding>utf-8</encoding>
  </configuration>
</plugin>
```

Spring 프레임워크 의존성 추가

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-jdbc</artifactId>  
  <version>3.2.3.RELEASE</version>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-web</artifactId>  
  <version>3.2.3.RELEASE</version>  
</dependency>
```

공통 값을 변수로 추출

```
<properties>  
  <org.springframework.version>3.2.3.RELEASE</org.springframework.version>  
</properties>
```

`${org.springframework.version}` 로 접근한다.

웹 자원 기본 디렉토리를 src/main/webapp에서 webapps로 변경

```
<plugin>  
  <artifactId>maven-war-plugin</artifactId>  
  <configuration>  
    <warSourceDirectory>webapp</warSourceDirectory>  
  </configuration>  
</plugin>
```


WTP에 배포할 수 있도록 설정

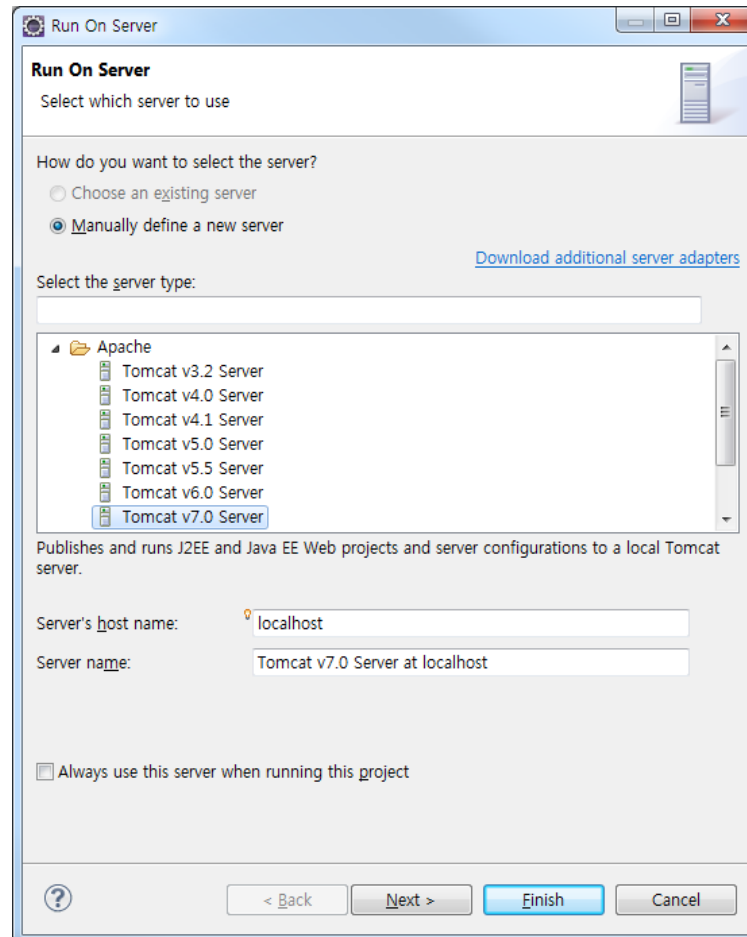
```
<plugin>
  <artifactId>maven-eclipse-plugin</artifactId>
  <version>2.6</version>
  <configuration>
    <wtpversion>2.0</wtpversion>
    <wtpContextName>/</wtpContextName>
  </configuration>
</plugin>
```

WTP(Web Tools Project)는 웹 프로젝트를 Eclipse에서 배포하고 테스트할 수 있도록 지원하는 플러그인

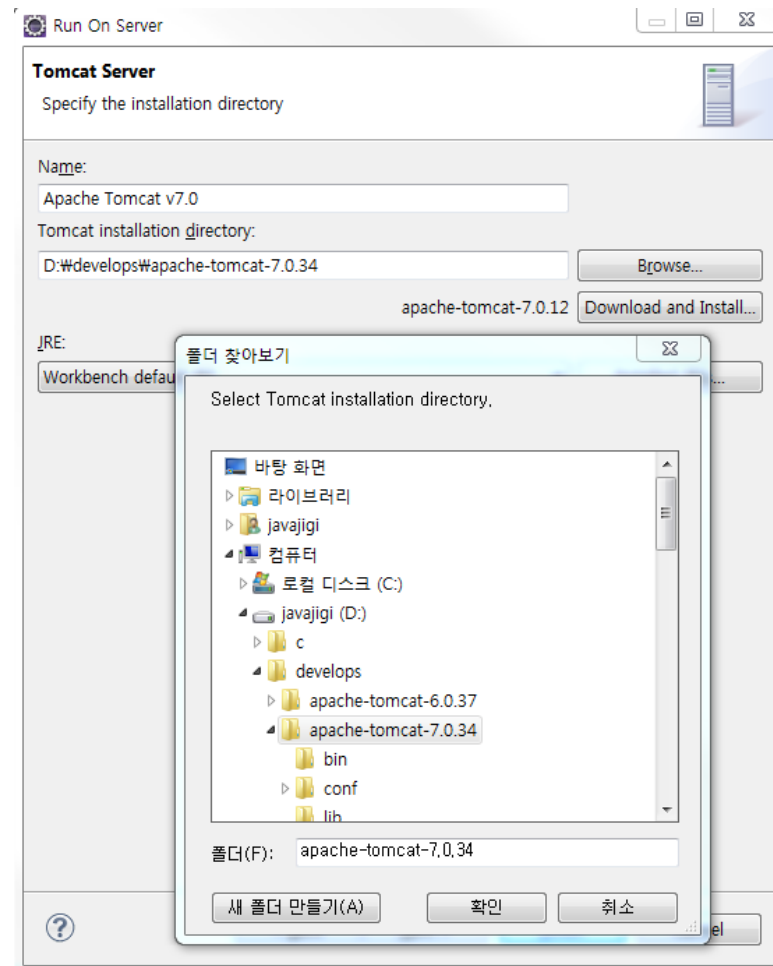
웹 개발환경 설정

—

- next-board 프로젝트 오른쪽 클릭 > Run As > Run On Server
- Apache > Tomcat v7.0 Server 선택 > Next

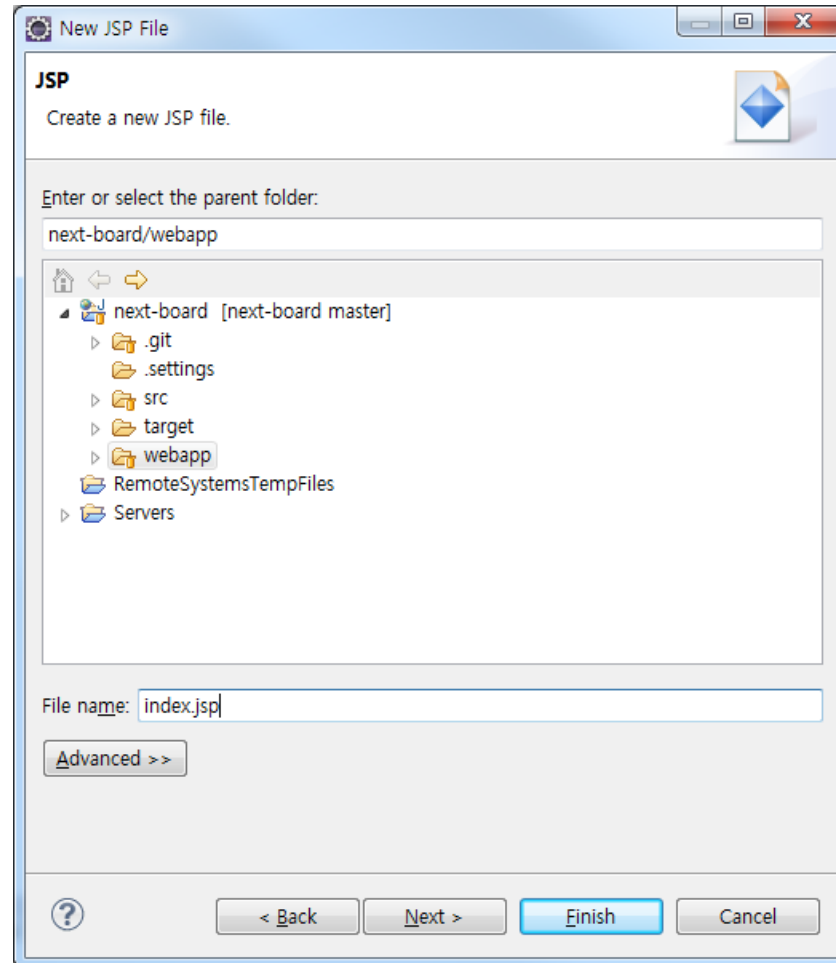


- Tomcat Installation directory의 Browse 버튼을 클릭해 Apache Tomcat 설치 디렉토리 선택 > Finish



- 브라우저에서 `http://localhost:8080/`로 접근한다.
- HTTP Status 404 - `/index.jsp` 에러 발생하면 설정 성공

- Cmd + N > jsp로 검색 > JSP File 선택 > next > next-board/webapp 디렉토리 선택 > File name을 index.jsp로 생성 > Finish



- index.jsp 파일의 <body></body>에 “안녕. NEXT”와 같은 메시지를 추가한 후 브라우저에서 <http://localhost:8080/>로 다시 접근한다.
- <body></body>에 추가한 메시지가 브라우저에 보이면 정상

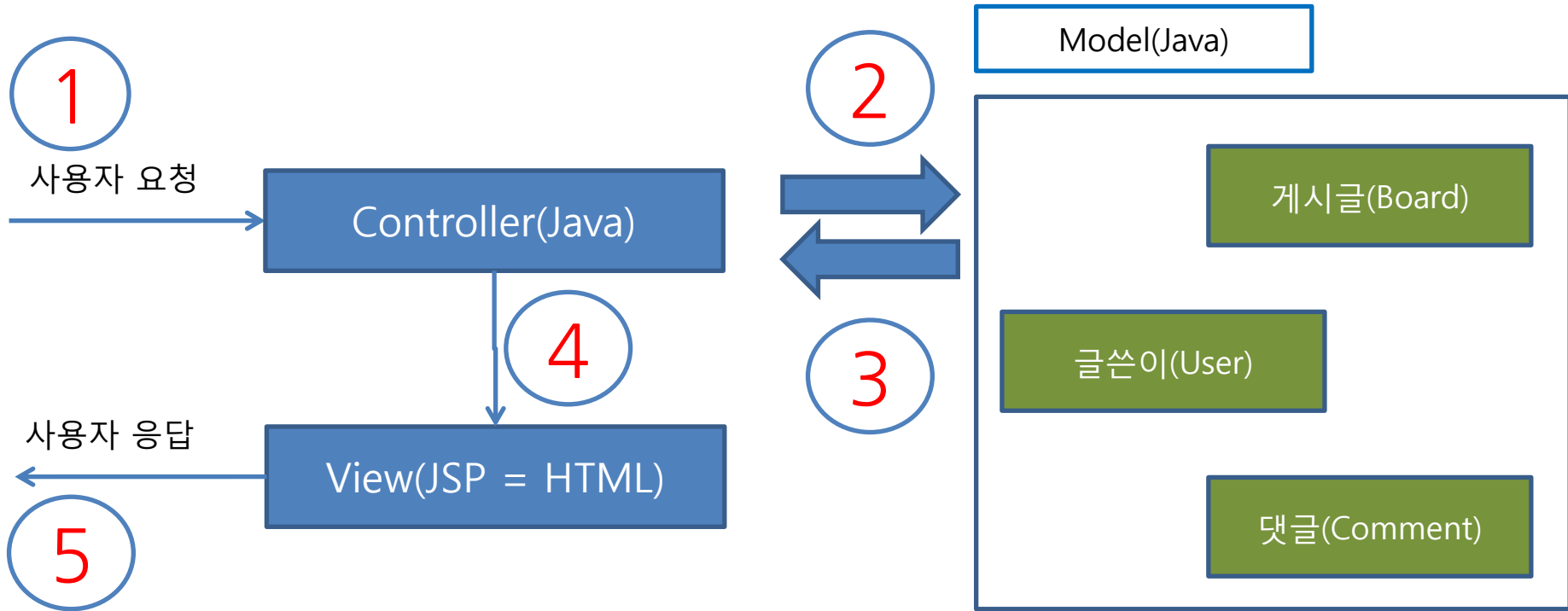
Eclipse와 Github, WTP 참고 문서

- 이클립스와 github 연동 :
<http://www.slipp.net/wiki/pages/viewpage.action?pageId=4161582>
- 이클립스에서 WTP 활용해 웹 프로젝트 배포
<http://www.slipp.net/wiki/pages/viewpage.action?pageId=5177373>

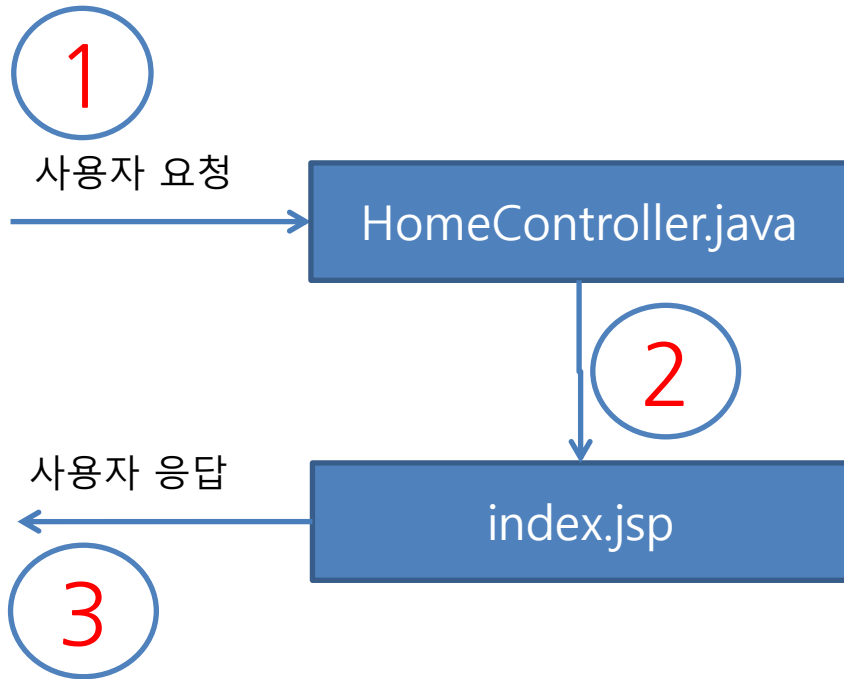
MVC (Model, View, Controller) 패턴

—

MVC



Controller + View 구조



<http://localhost:8080>

Controller

```
@Controller  
public class HomeController {  
    @RequestMapping("/")  
    public String index() {  
        return "index";  
    }  
}
```

View

```
// HomeController.java
```

```
package org.nhnnext.web;
```

```
@Controller
```

```
public class HomeController {
```

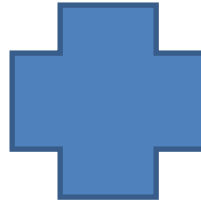
```
    @RequestMapping("/")
```

```
    public String index() {
```

```
        return "index";
```

```
    }
```

```
}
```



```
// webapp/WEB-INF/next-servlet.xml
```

```
<bean id="viewResolver"
```

```
    class="org.springframework.web.servlet.  
    view.InternalResourceViewResolver"
```

```
    p:prefix="/"
```

```
    p:suffix=".jsp" />
```

프로젝트의 웹 자원 디렉토리(webapp) + prefix 경로 + Controller 반환 경로 + suffix 경로
(next-board/webapp/index.jsp 파일을 View 파일로 찾는다.)

게시판 글쓰기 기능 구현

—

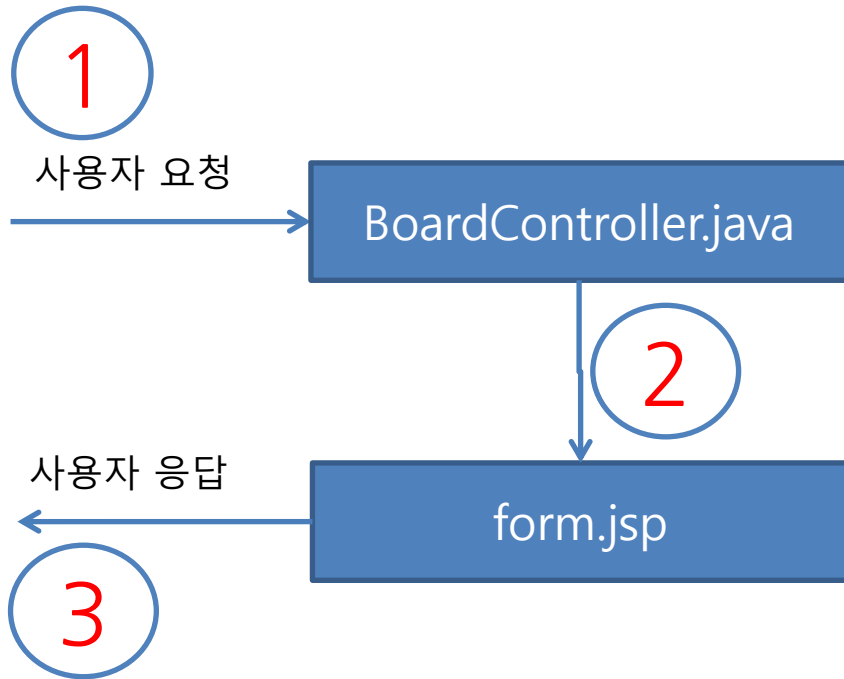
글쓰기 화면으로 이동

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>NEXT</title>
</head>
<body>
    <a href="/board/form">글쓰기</a>
</body>
</html>
```

Controller + View 추가

실습



<http://localhost:8080/board/form>

Controller

실습

```
// BoardController.java

package org.nhnnnext.web;

@Controller
public class BoardController {
    @RequestMapping("/board/form")
    public String form() {
        return "form";
    }
}
```

View

실습

사용자 입력페이지를 만들자
next-board/webapp/form.jsp

View 페이지 완성을 위한 단계

페이지내
콘텐츠 구조잡기

HTML

페이지에
스타일을 반영

CSS

페이지에
동적인 인터랙션
을 적용

JavaScript

구조

HTML은 웹 view 단의 구조를 완성한다.

뼈대를 완성하는 셈.

스타일을 변경하는 것은 CSS로 한다.(다음 주)



파이어폭스의 '파이어버그'나 크롬의 '개발자도구'는 UI개발과정에서 꼭 필요한 디버깅 도구이다

Sencha
Touch/HTML5/ExtJS
w3labs.kr/sencha
Sencha Touch/Architect 공식
리셀러, HTML5 패속 애플리케
이션 개발

AdChoices ▶

HTML Reference

HTML by Alphabet

HTML by Function

HTML Global Attributes

HTML Events

HTML Canvas

HTML Audio/Video

<oXygen/> DITA Editor

www.oxygenxml.com

Visually Edit DITA content Included DITA Open Toolkit

▶

SHARE THIS PAGE

f

t

e

m

a

i

s

좋아요 8.4만

WEB HOSTING

Best Web Hosting

eUK Web Hosting

UK Reseller Hosting

Domain, Hosting & Email

WEB BUILDING

HTML Reference - (HTML5 Compliant)

« W3Schools Home

Next Reference »

Ordered Alphabetically

New : New tags in HTML5.

Elements Resources Network Sources Timeline Profiles Audits Console

<!-- MainLeaderboard -->

▶<div id="mainLeaderboard">...</div>

<!-- TopRight -->

▶<div id="topRight">...</div>

<div style="clear:both;"></div>

▶<div id="leftcolumn">...</div>

▼<div id="main">

▶<h1>...</h1>

▶<div class="chapter">...</div>

<hr>

<h2>Ordered Alphabetically</h2>

▶<p>...</p>

▶<table class="reference notranslate">...</table>

▶<div class="chapter">...</div>

<hr>

<!-- SmallPS -->

▶<div id="div-gpt-ad-1379506098645-3" style="width:720px;margin-left:0px;">...</div>

<!-- LargePS -->

▶<div id="div-gpt-ad-1379506098645-1" style="width: 720px; margin-left:0px;">...</div>

<!-- BottomMediumRectangle -->

▶<div id="div-gpt-ad-1379506098645-0" style="width:300px; height:250px;">...</div>

▶<div id="err_form">...</div>

▶ Computed Style

▼ Styles

element.style {

}

Matched CSS Rules

#main {

width: 723px;

float: left;

margin: 7px;

margin-bottom: 0px;

overflow: hidden;

background-color: #ffffff;

}

div {

display: block;

}

Inherited from body

body, p, h1, h2, h3, h4, table, td, th, ul, ol, textarea, input {

font-family: verdana,helvetica,arial,sans-serif;

}

웹 페이지의 구조를 만드는 것은,
Word나 한글문서 만들기와도 비슷하다.

문서의 작성 순서가 자연스러워야 하며
의미에 맞는 태그를 사용해야 한다.

▶ 학습 내용 및 방법 Overview

* 수업 방식

한 주에 6시간 총 8주간 진행 됨. (48시간)

한 주에 수업은 두 번 진행하며, 2시간, 4시간 으로 나누어서 진행한다.

2시간 수업에서는 '뭘 배워야 하는지'에 대한 것을 '강의+실습'으로 진행하고,

4시간 수업에서는 '배운 걸 내 프로젝트에 어떻게 적용하지?'를 고민하고 자신의 프로젝트에 배운 부분을 개발하게 된다.

1) 수업방식 개요



2) 요일별 수업방식 상세

	화요일 2시간 수업	목요일 4시간 수업	수업이 없는 날
수업 성격	프로젝트 개발에 필요한 기초지식 배우기	자신의 프로젝트 개발	배운 내용을 내 프로젝트에 반영하기 위해서 추가적인 학습을 진행.
수업 방식	교수 강의, 학생 실습을 통한 이해	자신의 프로젝트 개발진행 학생 주도 학습. 교수의 지원받아 완성하기	자신의 프로젝트를 계속 개발하면서 필요한 걸 찾아 개발. 학습 진행 중 모르는 부분은 교수에게 도움 받기.
수업시간에 일어나는 것	강의자료를 통한 이해 교수와 질문/답변 배운 부분을 10~15분 단위로 실습	배운 내용을 자신의 프로젝트에 반영. 추가로 공부해야 할 것 찾아 반영. 진행 결과를 친구들에게 공유 및 토론 (수정 종료 20분전에 랜덤하게 지식공유를 목적으로 학생 발표)	

* 무엇을 어떤 순서로 배우는가?

Web UI개발은 'HTML,CSS,JavaScript' 세 가지 기술을 모두 이해하면서 개발해야 한다.

따라서 학생은 아래의 지식을 배우고 이를 실제 구현할 수 있어야 한다.

- 웹 사이트 UI의 구조 설계 <HTML>
- 디자인 스타일을 반영 <CSS>

HTML은 **다양한 의미를 표현**할 수 있도록 꽤 많은 TAG가 존재 함

Text

Link,

Image,

Tables,

Form,

Structure

<http://www.htmldog.com/reference/htmltags/>

UI 에 필요한 기술은 (HTML,CSS,JavaScript)
온라인을 통해서 참고,학습,테스트를 해볼 수 있음

다양한 HTML tag의 사용법을 알려주는 곳

<http://www.w3schools.com/html/default.asp>

<http://www.w3schools.com/tags/default.asp>

사용자의 입력을 받는 HTML Tag도 필요.

회원가입, 로그인, 검색창, 블로그 쓰기, 메일보내기..

제목

제목 색깔 ▾

☐ 제목 굵게

☐ 공지

☒ 댓글 허용

☒ 역인글 허용

☐ 알림

← →








형식 ▾

글꼴 ▾




크기 ▾

줄간격 ▾

B U I S C X^N X_N

 URL 

확장 컴포넌트

HTML 편집기

☐ 입력창 크기 자동조절

[파일 첨부](#) [선택 삭제](#) [본문 삽입](#)

문서 첨부 제한 : 0Byte/ 50.00MB

파일 제한 크기 : 50.00MB (허용 확장자 : *.*)



▶ [선택표\(.\)](#)를 이용하여 복수 등록

A login form with a light gray border. It contains two yellow text input fields. The first field contains the text "nx10019". The second field contains ten black dots, representing a masked password. Below the input fields is a blue button with the Korean text "로그인" (Login) in white.

```
<form>
```

```
  <input type="text" name="firstname">
```

```
  <input type="password" name="lastname">
```

```
</form>
```

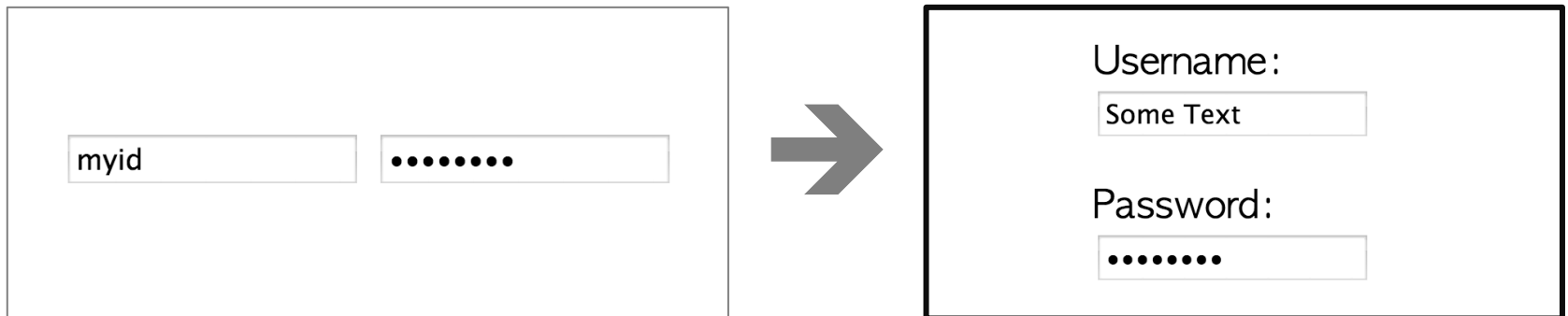
위 URL에서 FORM 양식을 확인한다.

<http://jsbin.com/IjIPuJo/1/edit>

미션

아래와 같은 UI화면으로 변경해본다.

‘label’이라는 태그가 무엇인지 찾아보고 활용해보자.



form.jsp를 새로 만들고 이곳에
제목과 내용을 입력 받을 수 있는 HTML Form 을 만들어보자.

우선 스타일과 배치는 신경쓰지 말자

글쓰기 화면

제 목:

사실상 우리가 이걸 듣는 이유는 뭐 잘 모르겠다는 거..
궁시렁 궁시렁..

올리기

Form 으로 입력한 데이터는 (제목, 본문)서버로 전달된다.

데이터 전달시 UI와 Server 간에 사전에 **약속된 이름**에 담아 전달되어야 한다

UI

title

프랑스를 다녀와서..

contents

이번 프랑스 여행은 어디를
가는가보다 누구랑 가는가
가 얼마나 중요한지 일깨우
준 기회였다.
정말 다음에는 혼자 여행을
가야 겠다는 생각이 들었다.



Server

```
title = get('title')  
contents = get('contents')
```

name 값을 지정된 이름으로 수정.

이제 서버에서는 form 데이터를 **title**과 **contents**라는 이름으로 받아서 확인할 수 있다.

UI

```
<input type="text" name="title">  
<textarea name="contents">
```

Server

```
@RequestMapping(value="/board", method=RequestMethod.POST)  
public String create(String title, String contents) {  
    System.out.println("title : " + title + " contents : " + contents);  
    return "redirect:/";  
}
```

View

실습

```
<form action="/board" method="post">  
  [...]   
  <input type="submit"/>  
</form>
```

Controller

실습

```
@Controller
public class BoardController {
    @RequestMapping("/board/form")
    public String form() {
        return "form";
    }

    @RequestMapping(value="/board", method=RequestMethod.POST)
    public String create(String title, String contents) {
        System.out.println("title : " + title + " contents : " + contents);
        return "redirect:/";
    }
}
```


궁금한 점

form.jsp 파일의 method가 뭘까?

왜 BoardController.create()는 method를 지정할까?

BoardController에서 method를 지정하지 않은 form()은 뭘지?

BoardController.create()의 반환 값에 "redirect:"는 뭘지?

BoardController 클래스를 보니 중복이 있다. 중복 제거할 수 없을까?

BoardController.create()에서 System.out.println() 사용해 출력만 하고 있다.

다음으로 무슨 작업을 해야 할까?

실습 시간 미션

—

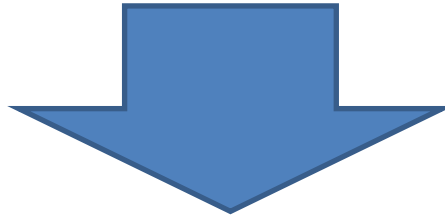


미션 1. 사진 웹 서비스를 위한 프로젝트 생성

미션 2. 사진 웹 서비스를 WTP에서 개발할 수 있는 환경 구축

미션 3. 사진을 추가할 수 있는 입력 화면(View)과 Controller 생성

미션 4. 앞의 궁금한 점 각각에 대해 학습하고, 공유하기



수업 중에 위 미션을 모두 완료하지 못하는 경우 과제로 진행한다.

실습의 경우 다음 수업 전까지 실습 완료 후 next dev 사이트나 github에 추가한 후 공유한다.

개인 학습 과제의 경우 학습한 내용을 카페에 제출한다.

궁금한 점

form.jsp 파일의 method가 뭘까?

왜 BoardController.create()는 method를 지정할까?

BoardController에서 method를 지정하지 않은 form()은 뭘지?

BoardController.create()의 반환 값에 "redirect:/"는 뭘지?

BoardController 클래스를 보니 중복이 있다. 중복 제거할 수 없을까?

BoardController.create()에서 System.out.println() 사용해 출력만 하고 있다.

다음으로 무슨 작업을 해야 할까?

사진 웹 서비스를 위한 프로젝트 시작

1. <https://github.com/javajigi/next-board> 에 접근한다.
2. 우측 상단의 Fork 버튼을 클릭해 next-board를 자신의 저장소로 만든다.
3. Fork한 저장소 이름을 원하는 이름으로 변경한다.
4. Eclipse에서 Git Repository Exploring Perspective로 이동한다.
5. Fork한 저장소를 추가한 후 checkout 한다.
6. Checkout한 프로젝트를 Eclipse에 import 한다.
7. pom.xml 파일을 열어 artifactId를 next-board에서 다른 이름으로 변경한다.
8. pom.xml 오른쪽 클릭 > Run As > Maven build ... > Goals에 eclipse:eclipse
입력 후 Run 실행
9. Import한 프로젝트에서 F5 실행(또는 오른쪽 클릭)해 Refresh