



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음

Chapter 03. JSP와의 첫 만남

목차

1. JSP 개요
2. JSP 처리 과정의 이해
3. JSP 프로그램 기술 변천
4. [기본실습] JSP프로그래밍 : Hello World JSP

학습목표

- JSP와 서블릿의 관계를 이해하고, JSP 프로그래밍과 관련된 기술을 알아본다.
- JSP 프로그램의 전체 처리 과정을 이해한다.
- JSP 프로그램 개발에 유리한 프로그래밍 모델을 알아본다.
- JSP 기술 변천 과정을 최근 소프트웨어 아키텍처 경향과 함께 살펴본다.
- 간단한 JSP 프로그램을 작성하고 실행하는 방법을 익힌다.

01. JSP 개요

1. 서블릿(Servlet)과 JSP(Java Server Page)

- 서블릿은 자바를 이용한 서버 프로그래밍 기술이다.
- 초기 웹 프로그래밍 기술인 CGI(Common Gateway Interface)를 대체하기 위해 개발되었으나, 느린 처리 속도, 많은 메모리 요구, 불편한 화면 제어 등의 한계로 PHP, ASP 등 서버 스크립트 언어 등장.
- JSP는 PHP와 유사한 형태로 HTML을 중심으로 자바 프로그램과의 유기적인 연결을 지원.

■ 서블릿 구현코드 예

```
01 public class HelloWorldServlet extends HttpServlet {
02     public void doGet(HttpServletRequest request,
03                         HttpServletResponse response)
04     throws ServletException, IOException {
05         response.setContentType("text/html; charset=EUC_KR");
06         PrintWriter out = response.getWriter();
07         out.println("<HTML> <HEAD> <TITLE>로그인
                                </TITLE> </HEAD>");
08         out.println("<BODY> <H2>Hello World : 헬로월드 </H2>");
09         out.println("오늘의 날짜와 시간은 : "+new
10                     java.util.Date());
11         out.println("</BODY> </HTML>");
12     }
13 }
```

■ JSP 구현코드 예

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <HTML>
03 <HEAD> <TITLE>Hello World</TITLE> </HEAD>
04 <BODY> <H2>Hello World : 헬로월드 </H2>
05   오늘의 날짜와 시간은 : <%= new java.util.Date() %>
06 </BODY>
07 </HTML>
```

01. JSP 개요

■ JSP의 특징

- ❶ 자바의 모든 기능을 사용할 수 있어 발전 가능성이 무한하다.
- ❷ 서블릿으로 컴파일된 후 메모리에서 처리되기 때문에 많은 사용자의 접속도 원활하게 처리할 수 있다.
- ❸ JSP 또는 다른 서블릿 간의 데이터를 쉽게 공유 할 수 있다.
- ❹ 빈즈(Beans)라고 하는 자바 컴포넌트를 사용할 수 있다.
- ❺ 커스텀 태그를 만들어 사용할 수 있으며, JSTL(JSP Standard Tag Library)과 같은 태그 라이브러리를 이용할 수 있다.
- ❻ 스트러츠, 스프링 @MVC 등 다양한 프레임워크와 결합하여 개발할 수 있다.

2. JSP 학습에 필요한 기술

- JSP는 웹 프로그래밍 기술로 HTML, 자바스크립트, CSS와 같은 기본 웹 프로그래밍 경험이 요구됨.

[표 3-1] 웹 프로그래밍을 위한 기본 기술 경험의 요구 수준

기술	설명	요구 수준
HTML	클라이언트 기술로서, 웹 프로그램의 기본이 되며 시각적인 부분을 담당한다.	• HTML 문서 구조와 기본 태그 • FORM 관련 태그 • HTML5 기본 구조
자바스크립트	웹 화면과 사용자와의 상호작용 및 동적 웹 페이지를 구현할 때 필요한 기술이다.	• 기본 문법 • 객체와 메서드 • 내장 객체 • 이벤트 핸들링
CSS	웹 화면의 레이아웃과 디자인 요소를 구현할 때 필요한 기술이다.	• 스타일시트 정의 및 셀렉터 이해 • DOM 연동에 의한 동적 스타일 제어

01. JSP 개요

- JSP는 자바언어 기반이며 개발 시 순수 자바 코드가 50% 이상으로 탄탄한 자바 기본기가 요구됨.

[표 3-2] 자바 관련 기본 기술 경험의 요구 수준

기술	설명	요구 수준
자바	소스코드를 작성하기 위한 프로그래밍 기본 언어로서, Java SE를 기준으로 한다.	<ul style="list-style-type: none">• 자바 기본• 상속, 오버로딩, 오버라이딩• java.util, java.io 패키지• 예외 핸들링• 객체지향 개념• 인터페이스 구현• 스레드
JDBC	Java DataBase Connectivity의 약자로서, 자바에서 데이터베이스 프로그래밍을 하기 위한 기술이다.	<ul style="list-style-type: none">• JDBC 드라이버 세팅• PreparedStatement• 기초 SQL문• ResultSet• 데이터 핸들링
서블릿	JSP의 기본이 되는 자바 기반의 웹 프로그래밍 핵심 기술이다.	<ul style="list-style-type: none">• 서블릿 구조 이해• request, response 처리• 간단한 서블릿 프로그래밍• GET/POST 처리

01. JSP 개요

- 이외 추가적으로 다음 기술들에 대한 경험이 있다면 고급 웹 프로그래밍 학습에 도움이 됨.

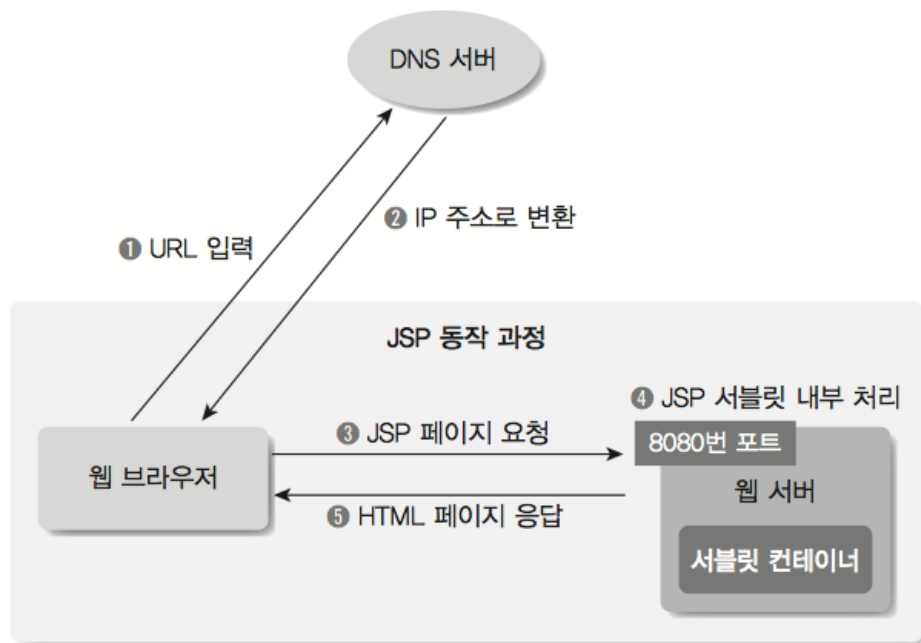
[표 3-3] 고급 웹 프로그래밍을 위한 주변 기술 경험의 요구 수준

기술	필요성	요구 수준
데이터베이스	프로그램의 데이터를 처리하려고 할 때 반드시 필요하다.	<ul style="list-style-type: none">• 다양한 SQL문의 사용• 데이터베이스 연계 프로그래밍 경험• 데이터베이스 함수 및 내장 프로시저
XML	eXtensible Markup Language의 약자로서, 확장 가능한 구조적 문서 표현을 제공한다. 많은 프로그램에서 데이터 구조를 XML 기반으로 처리한다.	<ul style="list-style-type: none">• XML 스키마 및 DTD 이해• XML DOM 개요
모바일 프로그래밍	최근에는 스마트폰을 중심으로 하는 모바일 기반의 개발이 증가하고 있는 추세다.	<ul style="list-style-type: none">• 안드로이드 혹은 아이폰 앱 개발 경험• 하이브리드 앱 개발 경험
프레임워크	개발자로 하여금 더욱 좋은 프로그램을 만들 수 있도록 미리 제공되는 틀을 말한다.	<ul style="list-style-type: none">• 소프트웨어 아키텍처 이해• 스프링 프레임워크• 스프링3 @MVC

02. JSP 처리 과정의 이해

1. JSP 전체 동작 과정

- JSP 는 HTML 과 유사한 처리 과정을 거치나 HTML이 단순 서버 파일을 브라우저로 보내주는 것에 비해 JSP는 서버에서 프로그램이 실행된 결과를 웹 브라우저로 전달하는 차이가 있음.



[그림 3-1] JSP 전체 동작 과정

02. JSP 처리 과정의 이해

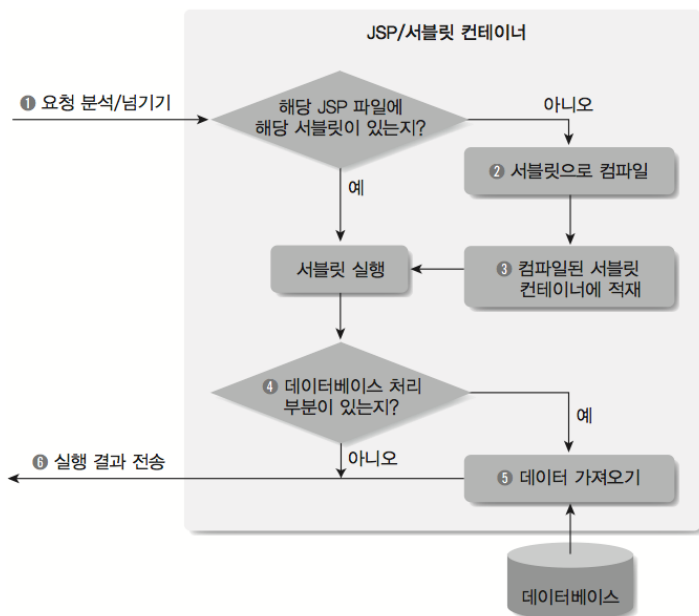
2. 서블릿 컨테이너 내부 과정

▪ JSP와 서블릿 차이

- JSP는 HTML과 같은 일반적인 텍스트 파일 구조
- 서블릿은 자바 소스로 작성된 클래스 파일 구조
- JSP는 서블릿 컨테이너에 의해 서블릿 형태의 자바 소스로 변환되어 클래스로 컴파일 됨

▪ 서블릿 컨테이너

- 서블릿 컨테이너는 서블릿을 실행하고 JSP를 서블릿 코드로 변환하는 기능을 수행함.
- 변환된 JSP의 서블릿 클래스를 실행하고 웹 서버의 메모리에 적재하고 사용자 요청에 따라 실행.

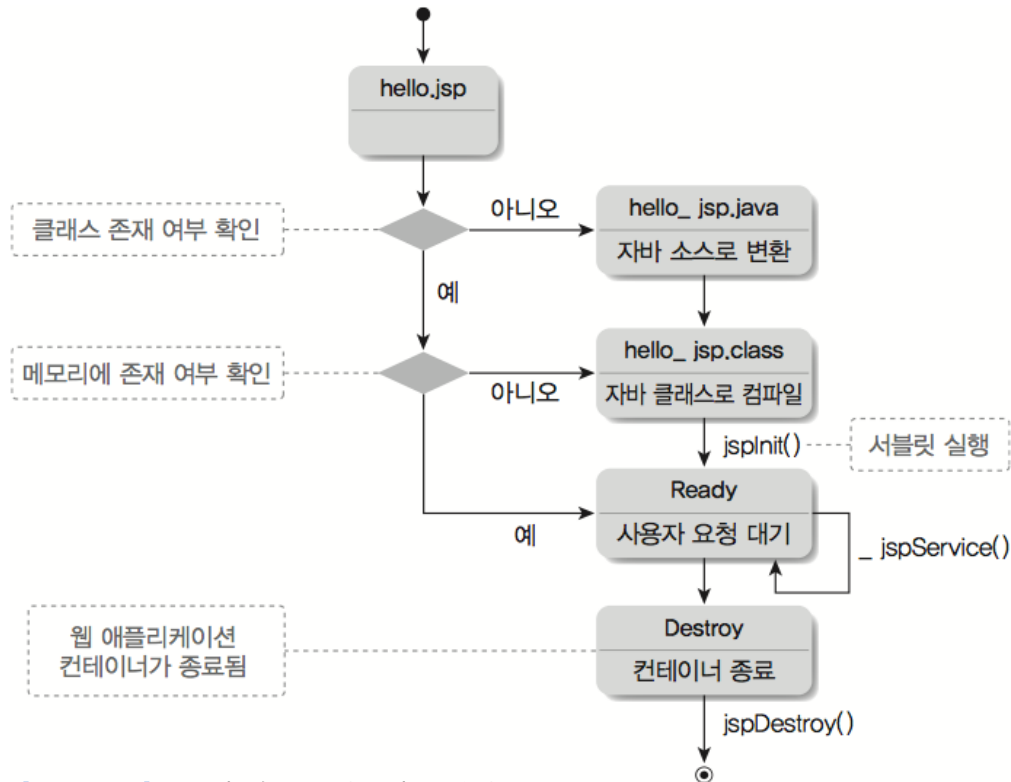


- ① 웹 서버로부터 JSP에 대한 사용자 요청이 컨테이너로 전달된다.
- ② 요청 JSP에 대한 서블릿이 존재하면 다음 단계로 진행하고, 존재하지 않을 경우 JSP를 .java 파일로 변환한 다음 .class 파일로 컴파일한다.
- ③ 컴파일된 서블릿 클래스를 컨테이너의 메모리에 적재하고 실행한다.
- ④~⑤ 데이터베이스 처리 혹은 별도의 기능을 위한 클래스 호출 등이 있다면 실행하고 결과를 취합해 HTML 형태로 구성한다.
- ⑥ HTML 형태의 결과를 웹 서버를 경유해 사용자 브라우저에 전달한다.

[그림 3-2] JSP 서블릿 컴파일과 처리 과정

02. JSP 처리 과정의 이해

- JSP 파일은 일반 텍스트를 비롯해 HTML 코드와 특수태그, 자바 코드가 섞여 있다.
- 서블릿으로 변환된 JSP는 컨테이너에 의해 생명주기가 관리된다.
- 서블릿으로 변환된 JSP는 `jspInit()` 메서드에 의해 실행되고 Ready 상태가 되며 이후 사용자 요청은 `_jspService()` 메서드가 쓰레드 형태로 호출되어 실행된다.
- 컨테이너에 의해 JSP 서블릿이 종료될 때에는 `jspDestroy()` 메서드가 실행된다.



[그림 3-3] JSP와 서블릿 변환과 상태변화

02. JSP 처리 과정의 이해

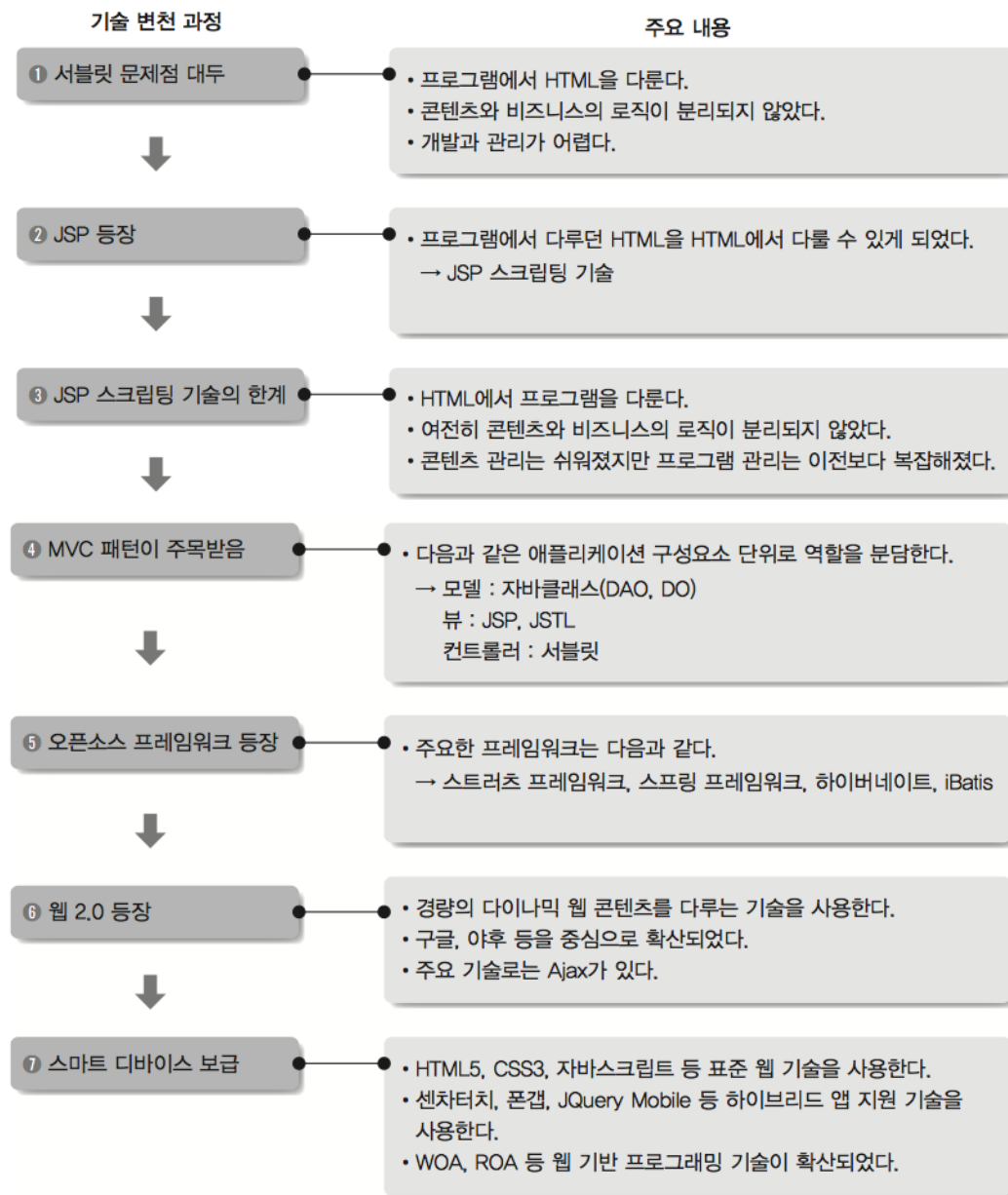
■ JSP에 관해서 이것만은 알고 있자.

- ❶ JSP는 일반 텍스트 파일로 되어 있다(텍스트 파일은 컴퓨터가 이해할 수 없다. 즉 실행 가능한 프로그램이 아니며 특정 동작을 할 수 없다).
- ❷ JSP는 HTML 코드와 몇몇 특수한 태그, 그리고 자바 코드가 섞여 있다.
- ❸ 사용자가 요청할 경우 JSP는 컨테이너(톰캣)에 의해 서블릿 형태의 .java 소스로 변환되고 컴파일된다.
- ❹ 컴파일된 .class는 컴퓨터에서 실행할 수 있는 형태로 특정한 기능을 수행할 수 있게 된다. 이후 소스 변경 전까지 해당 파일은 메모리에 상주하면서 다시 컴파일 되지 않고 서비스된다.

03. JSP 프로그램 기술 변천

1. 서블릿과 JSP 기술 변천

- 서블릿 : 일반 자바 프로그램을 통해 웹 콘텐츠를 표현함.
- JSP : HTML 기반으로(즉 화면을 중심으로) 프로그램과의 연동을 지원함.
- JSP와 HTML 코드의 혼용으로 인한 문제점 해결을 위한 MVC 패턴 등장
- 개발 생산성과 확장성 등 개발 수준 향상을 위한 오픈소스 프레임워크 등장
- 새로운 웹 기술과 새로운 디바이스의 보급 확산



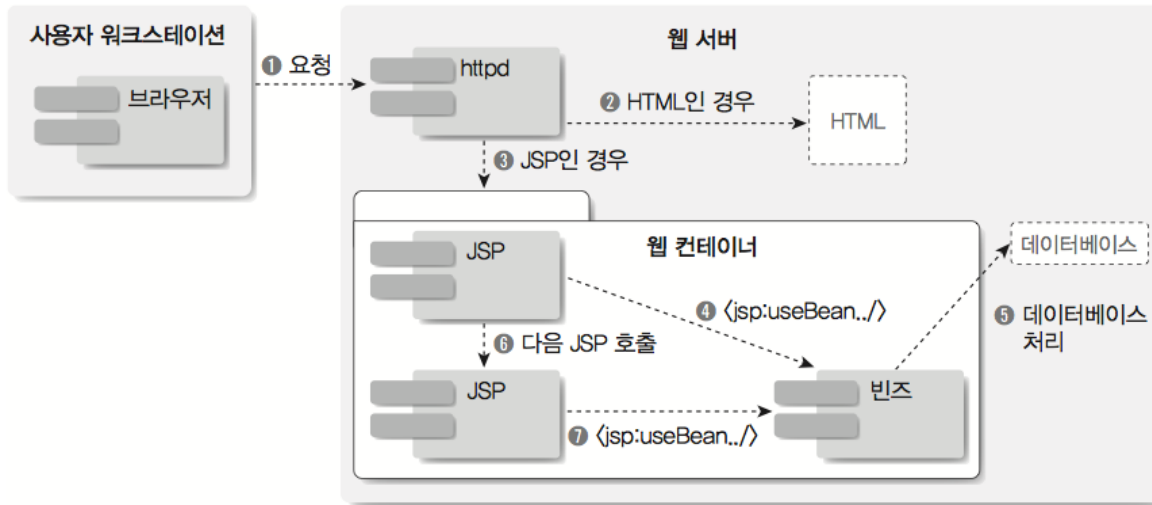
[그림 3-4] 서블릿과 JSP의 기술변천

2. JSP 프로그램 모델 유형 비교

- 프로그램 모델은 구조와 관련된 것으로 실행결과는 동일하나 프로그램 개발 및 유지보수에는 많은 차이가 있을 수 있음.
- 유형-1 : 계산기 프로그램(calc.jsp) – [교재 p.88 참고](#)
 - 빈즈 클래스를 사용하지 않고 스크립트릿만을 사용하는 방법
- 유형-2 : 계산기 프로그램(calc.jsp) – [교재 p.90 참고](#)
 - 빈즈 클래스와 `<jsp:useBean>` 액션을 적극적으로 사용하는 방법

03. JSP 프로그램 기술 변천

■ 빈즈를 이용한 JSP 처리 과정



[그림 3-5] 빈즈를 사용한 JSP 처리 과정

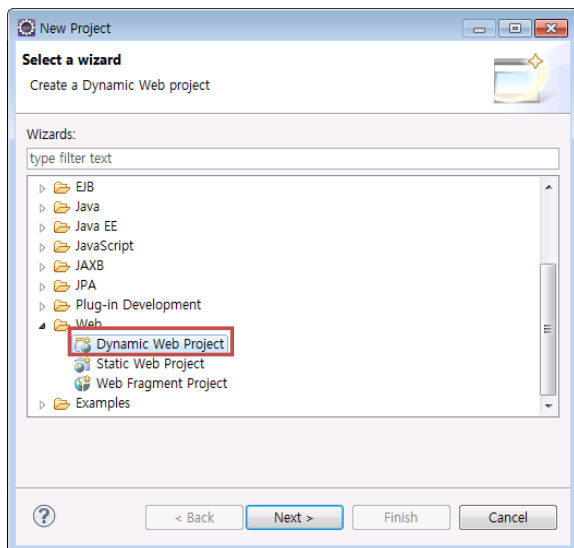
- 1 웹 브라우저 요청에 대해서 웹 서버가 http 프로토콜에 따라 사용자 요청을 처리한다.
- 2 ~ 3 웹 서버는 HTML과 JSP인 경우를 구분해서 처리한다.
- 4 JSP의 경우 useBean 액션을 이용해서 빈즈 클래스와 연동한다.
- 5 필요한 경우 데이터베이스 관련 작업 등을 처리한다.
- 6 처리한 결과를 보여주는 JSP를 호출한다.
- 7 필요한 경우 결과를 보여주는 JSP에서도 빈즈를 사용할 수도 있다.

04. [기본실습]JSP 프로그래밍: Hello World JSP

이클립스를 통한 JSP 개발 과정과 JSP 기본 구조를 간단한 프로그램 개발을 통해 알아본다.

1. 이클립스 프로젝트 생성

- 이클립스를 실행하고 [File] → [New] → [Project]를 선택하면 미리 정의된 특정 유형의 프로젝트 템플릿을 이용하여 프로젝트를 생성할 수 있다.
- 다이나믹 웹 프로젝트 생성하기
 - 트리 메뉴 중 [Web]을 선택하고 [Dynamic Web Project]를 선택하면 JSP 개발을 위한 프로젝트가 생성된다.

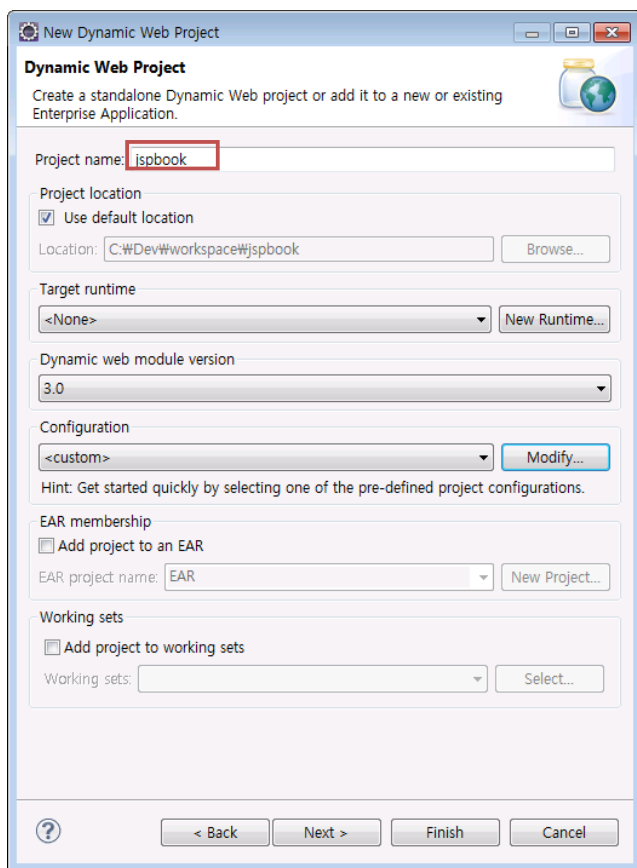


[그림 3-6] 다이나믹 웹 프로젝트 생성

04. [기본실습]JSP 프로그래밍: Hello World JSP

■ 기본 정보 설정하기

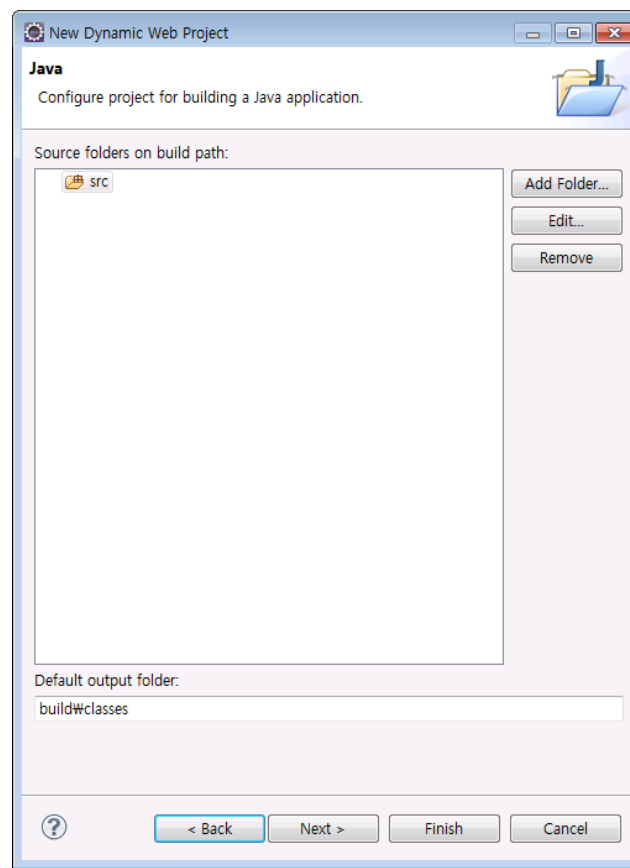
- 프로젝트 이름, 웹 모듈 버전 등 프로젝트 기본 정보를 설정한다.



[그림 3-7] 프로젝트 기본 정보 설정

■ 소스 폴더 설정하기

- 자바 클래스 소스 폴더를 지정한다.

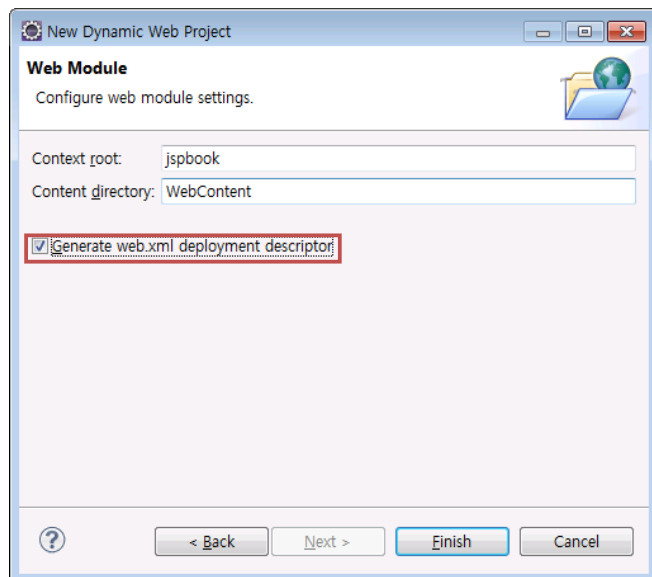


[그림 3-8] 소스 폴더 설정

04. [기본실습]JSP 프로그래밍: Hello World JSP

■ 웹 모듈 설정하기

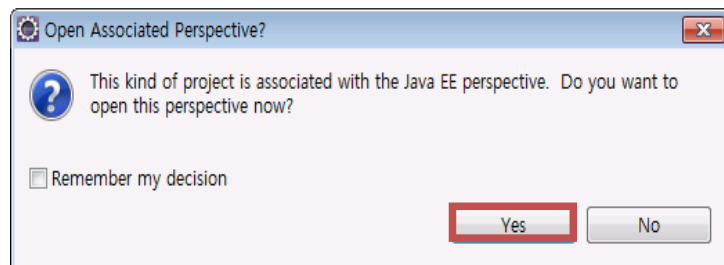
- Context Root
 - 웹 애플리케이션의 메인 접속 경로를 말함.
 - <http://localhost:8080/jspbook> 과 같이 JSP 실행을 위한 기본 URL에 적용됨.
- Content Directory
 - JSP, HTML, 이미지 등 기본 웹 콘텐츠가 위치하는 디렉터리.
 - 이클립스 프로젝트 구조에서는 [WebContent] 폴더가 기본 값으로 사용됨.



[그림 3-9] 웹 모듈 설정

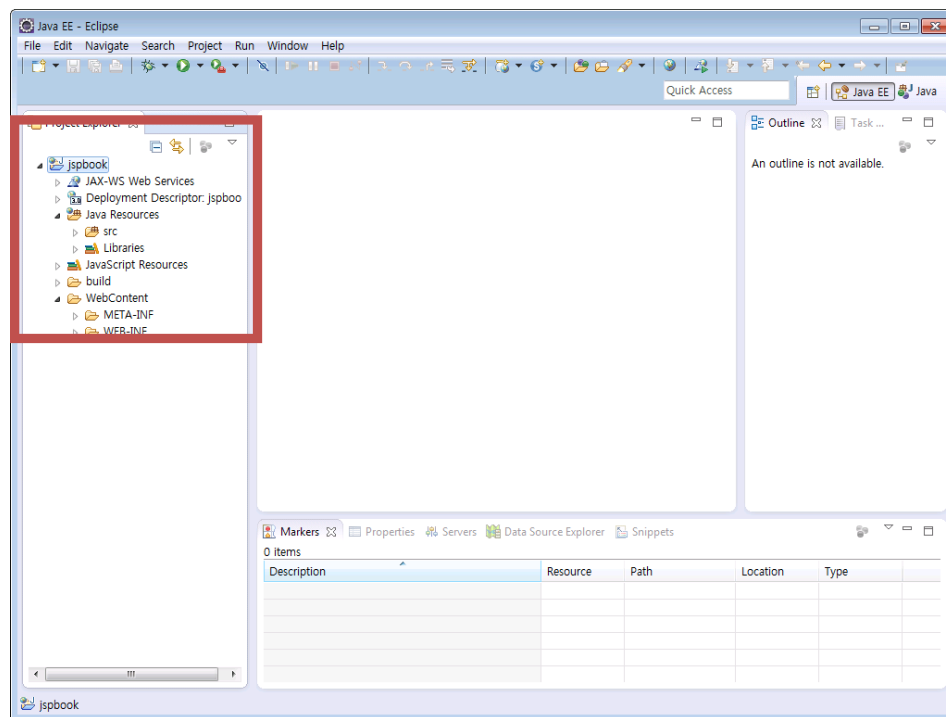
04. [기본실습]JSP 프로그래밍: Hello World JSP

■ 퍼스펙티브 변경하기



[그림 3-10] 퍼스펙티브 변경

■ 생성된 프로젝트 확인

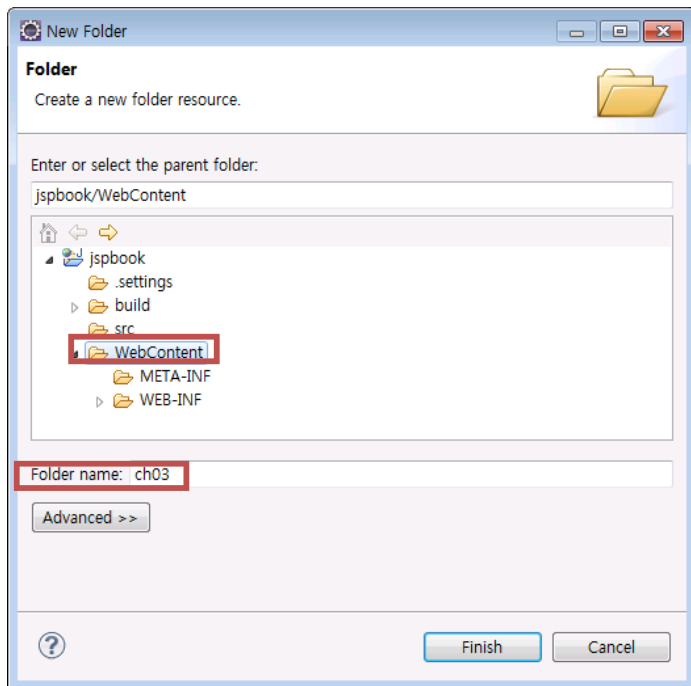


[그림 3-11] 생성된 프로젝트 확인

04. [기본실습]JSP 프로그래밍: Hello World JSP

2. Hello World 프로그램 소스 작성

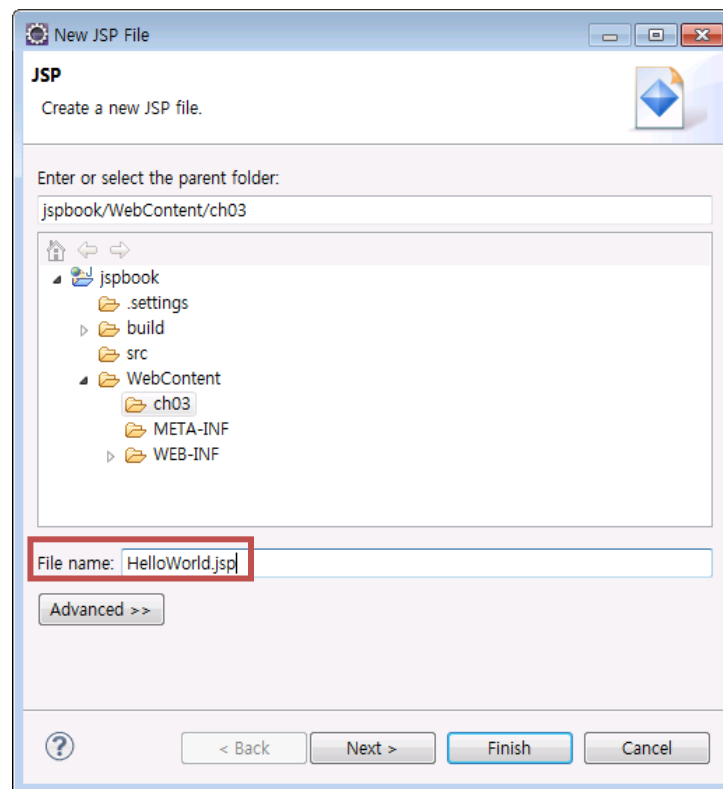
폴더 생성



[그림 3-12] ch03 폴더 생성

JSP 생성

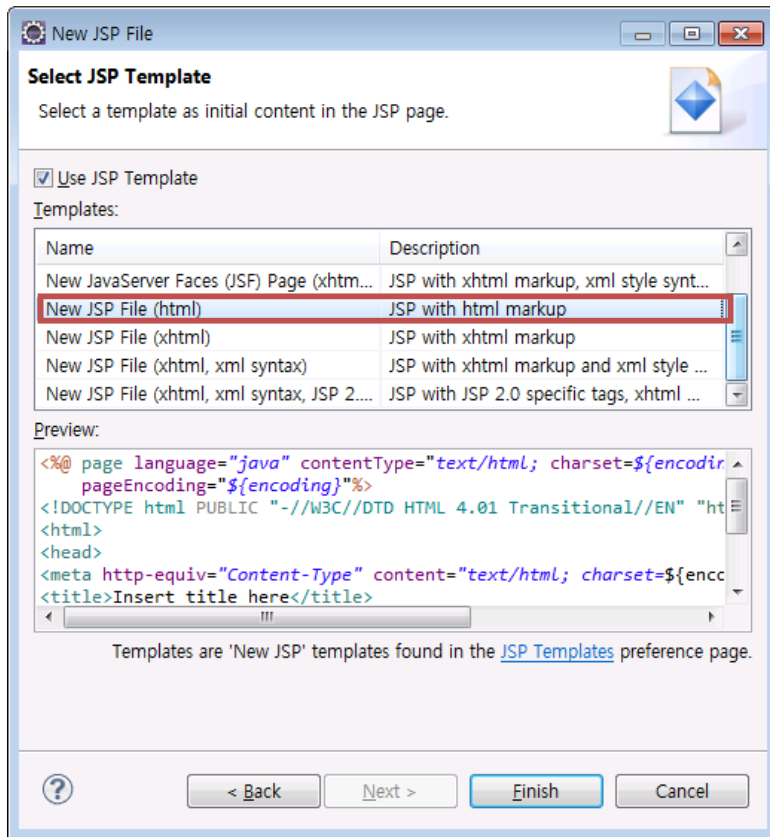
① JSP 파일 이름 지정하기



[그림 3-13] JSP 파일 이름 지정

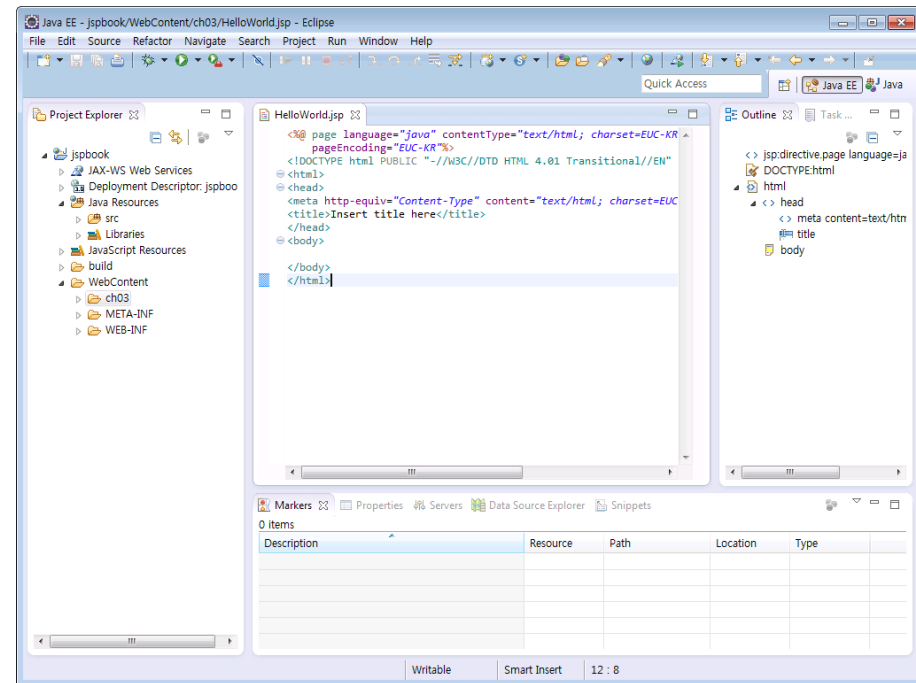
04. [기본실습]JSP 프로그래밍: Hello World JSP

② 템플릿 코드 지정하기



[그림 3-14] 템플릿 선택

③ 생성된 코드 확인하기



[그림 3-15] 생성된 기본 코드

04. [기본실습] JSP 프로그래밍: Hello World JSP

■ 소스코드 작성 : 헬로월드(HelloWorld.jsp)

- **page 지시어** : 모든 jsp 파일에 기술되어야 하는 요소로 현재 jsp 문서와 관련된 처리 정보를 기술함.

```
01 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

- **HTML 선언부 및 기본 태그**

```
02 <!DOCTYPE html PUBLIC "-//W3c//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
03 <HTML>
04 <HEAD>
05 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
06 <TITLE>HelloWorld</TITLE>
07 </HEAD>
08 <BODY>
```

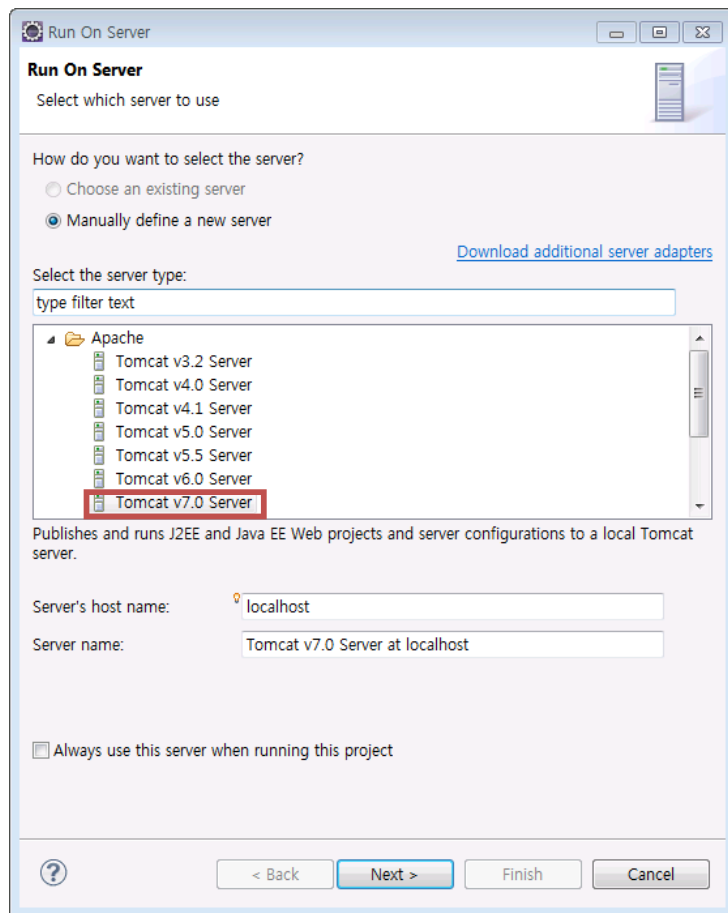
- **JSP 표현식** : 자바 코드를 이용해 간단한 출력을 위해 사용. 여기서는 현재 날짜와 시간 정보가 출력됨.

```
12 현재 날짜와 시간은 : <%=new java.util.Date() %>
```

04. [기본실습]JSP 프로그래밍: Hello World JSP

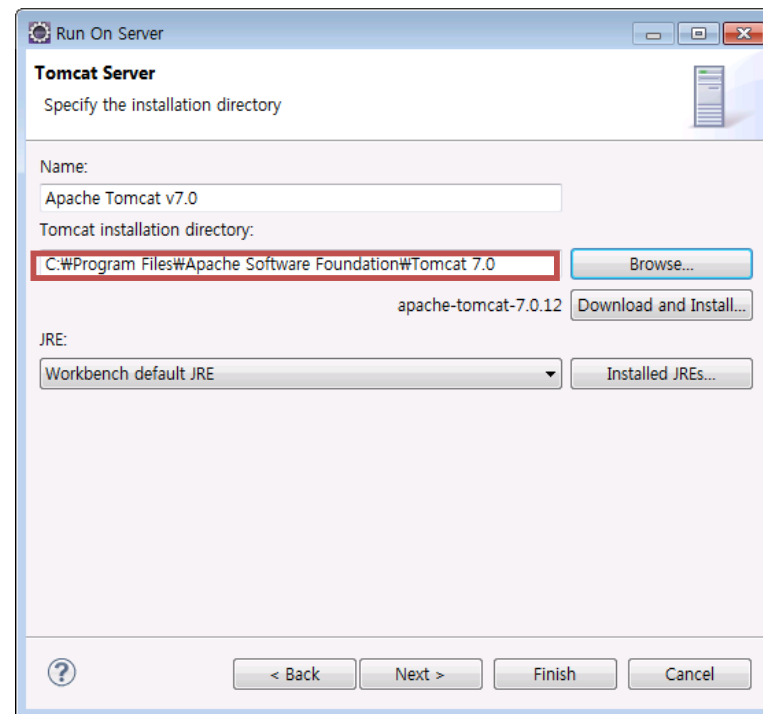
3. 서버 설정 및 실행

■ 서버 설정하기



[그림 3-16] 서버 설정

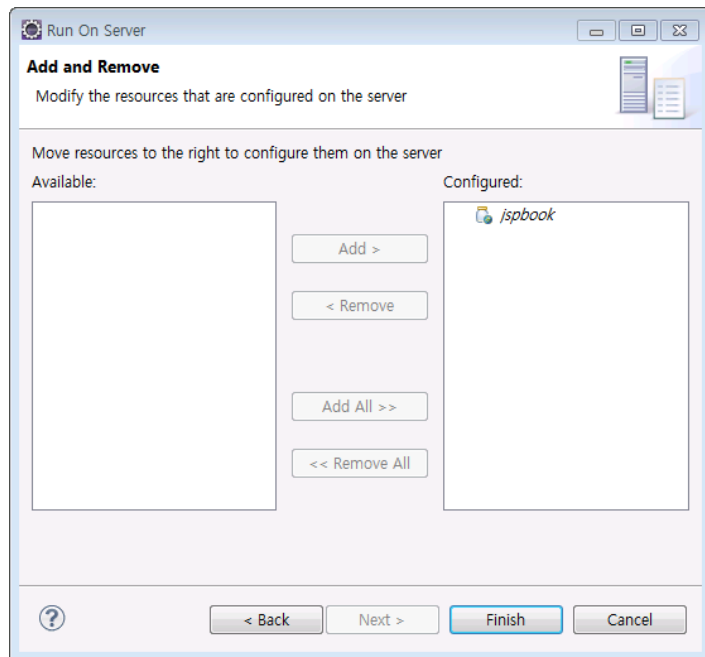
■ 톰캣 폴더 지정하기



[그림 3-17] 톰캣 폴더 지정

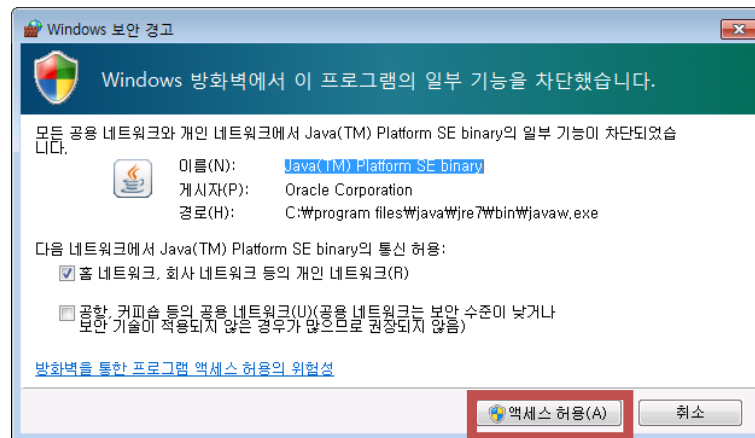
04. [기본실습]JSP 프로그래밍: Hello World JSP

■ 실행할 프로젝트 선택하기



[그림 3-18] 실행 프로젝트 선택

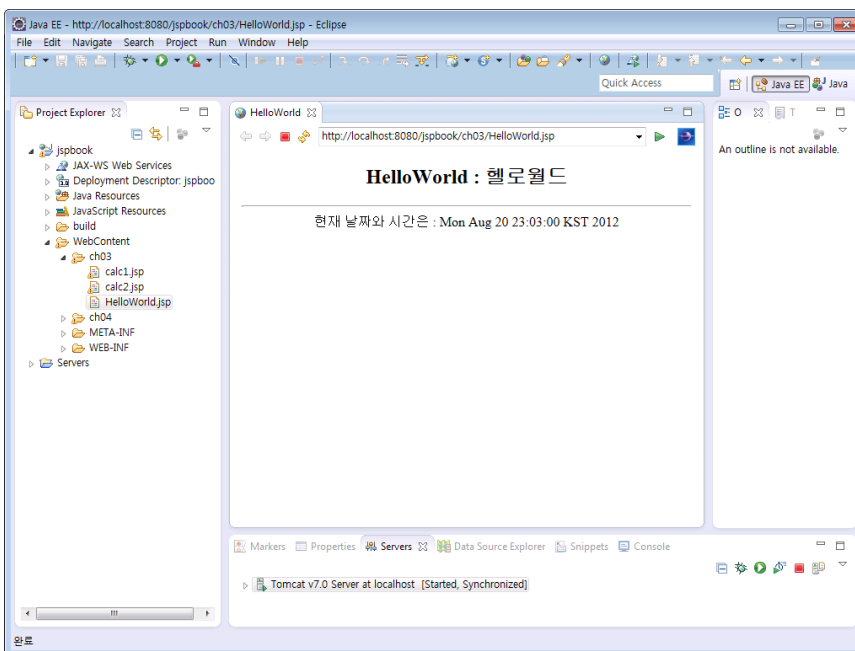
■ 보안 경고 해제하기



[그림 3-19] 윈도우 보안 경고

04. [기본실습]JSP 프로그래밍: Hello World JSP

■ 실행 결과 확인하기



[그림 3-20] 실행 결과 확인

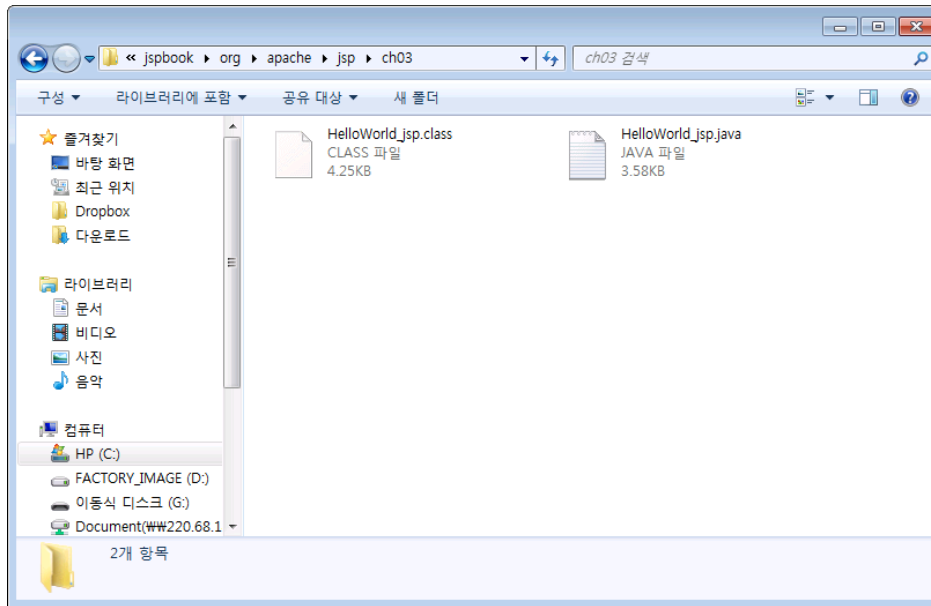


[그림 3-22] 외부 브라우저(크롬)를 이용한 실행 결과

04. [기본실습]JSP 프로그래밍: Hello World JSP

4. 서블릿으로 변환된 소스 확인

- [c:\wdev\workspace\metadata\plugins\org.eclipse.wst.server.core\wtmp0\work\Catalina\localhost\jspbook\org\apache\jsp\ch0] 폴더에 위치



[그림 3-23] JSP가 변환된 자바 파일이 있는 폴더



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음