

JSP 웹 프로그래밍

1 JSP 개요

Contents

1

웹과 JSP 프로그래밍 이해하기

2

JSP 개발 환경 구축하기

3

[웹 쇼핑몰] 프로젝트 생성하기

학습목표

- 웹의 개념과 동작 원리를 이해합니다.
- 웹 프로그래밍을 위한 JSP의 개념과 특징을 이해합니다.
- JSP 프로그래밍에 필요한 개발 환경을 구축합니다.
- 동적 웹 프로젝트를 생성하여 간단한 프로그램을 실행합니다.
- 웹 쇼핑몰 시작 페이지를 만들어봅니다.

1. 웹과 JSP 프로그래밍 이해하기

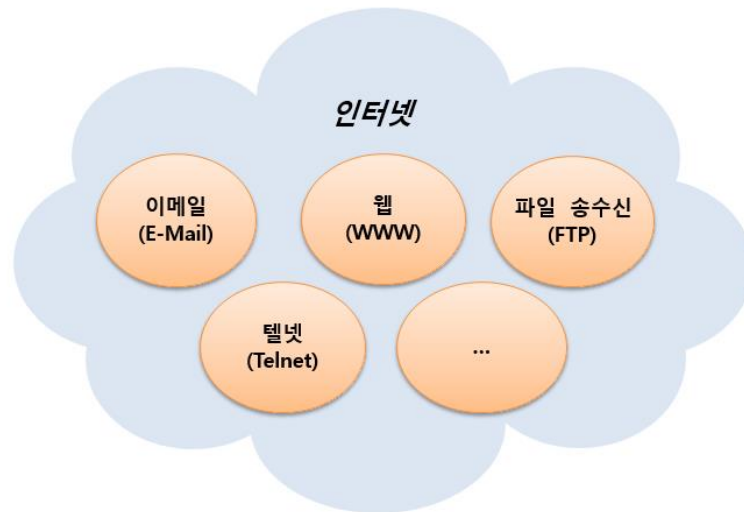
❖ 인터넷과 웹의 개요

■ 인터넷

- 컴퓨터가 서로 연결되어 TCP/IP라는 통신 프로토콜을 이용하여 정보를 주고받는 전 세계의 컴퓨터 네트워크

■ 웹

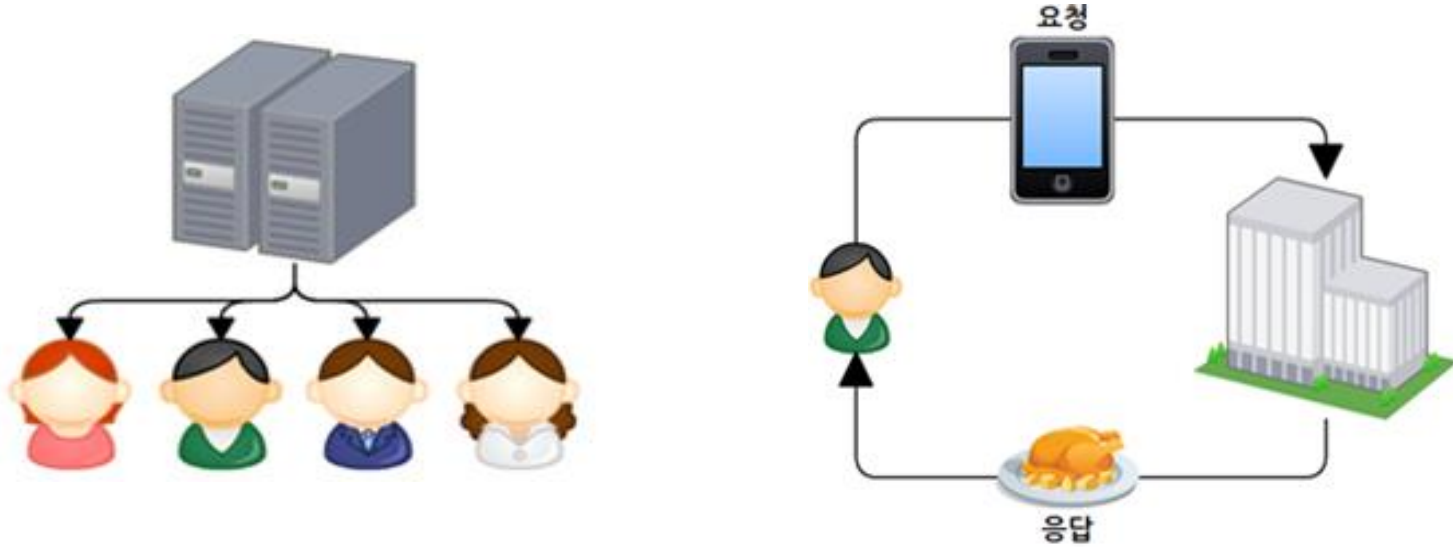
- 인터넷에 연결된 컴퓨터들을 통해 사람들이 정보를 공유할 수 있는 정보 공간
- 월드 와이드 웹(world wide web)의 줄임말



1. 웹과 JSP 프로그래밍 이해하기

❖ 웹의 동작 원리

- 웹은 기본적으로 클라이언트/서버 방식으로 동작



- 가장 널리 쓰이는 웹 서버
 - 아파치(Apache), 톰캣(Tomcat), IIS(Internet Information Server)

1. 웹과 JSP 프로그래밍 이해하기

❖ 정적 웹 페이지와 동적 웹 페이지

■ 정적 웹 페이지

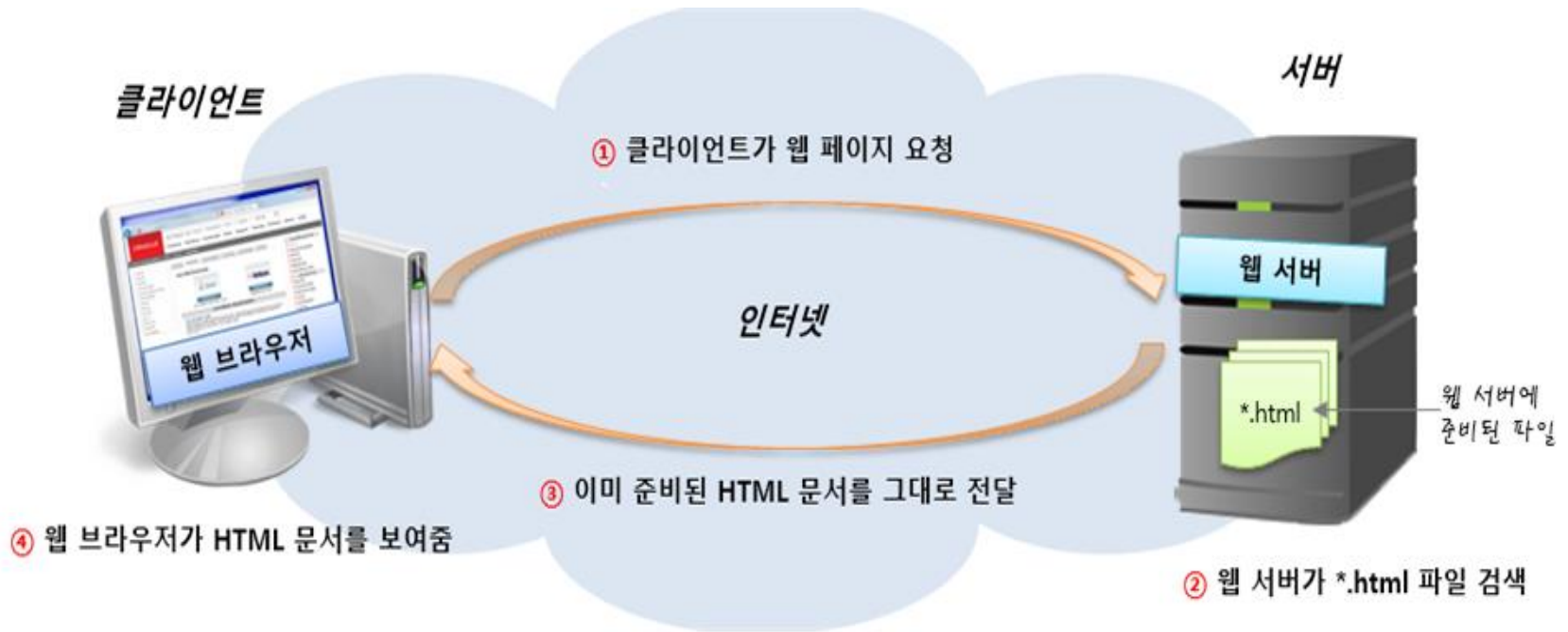
- 컴퓨터에 저장된 텍스트 파일을 그대로 보는 것
- HTML(HyperText Markup Language)

■ 동적 웹 페이지

- 저장된 내용을 다른 변수로 가공 처리하여 보는 것
- PHP(Personal Home Page), ASP(Active Server Page), JSP(Java Server Page)

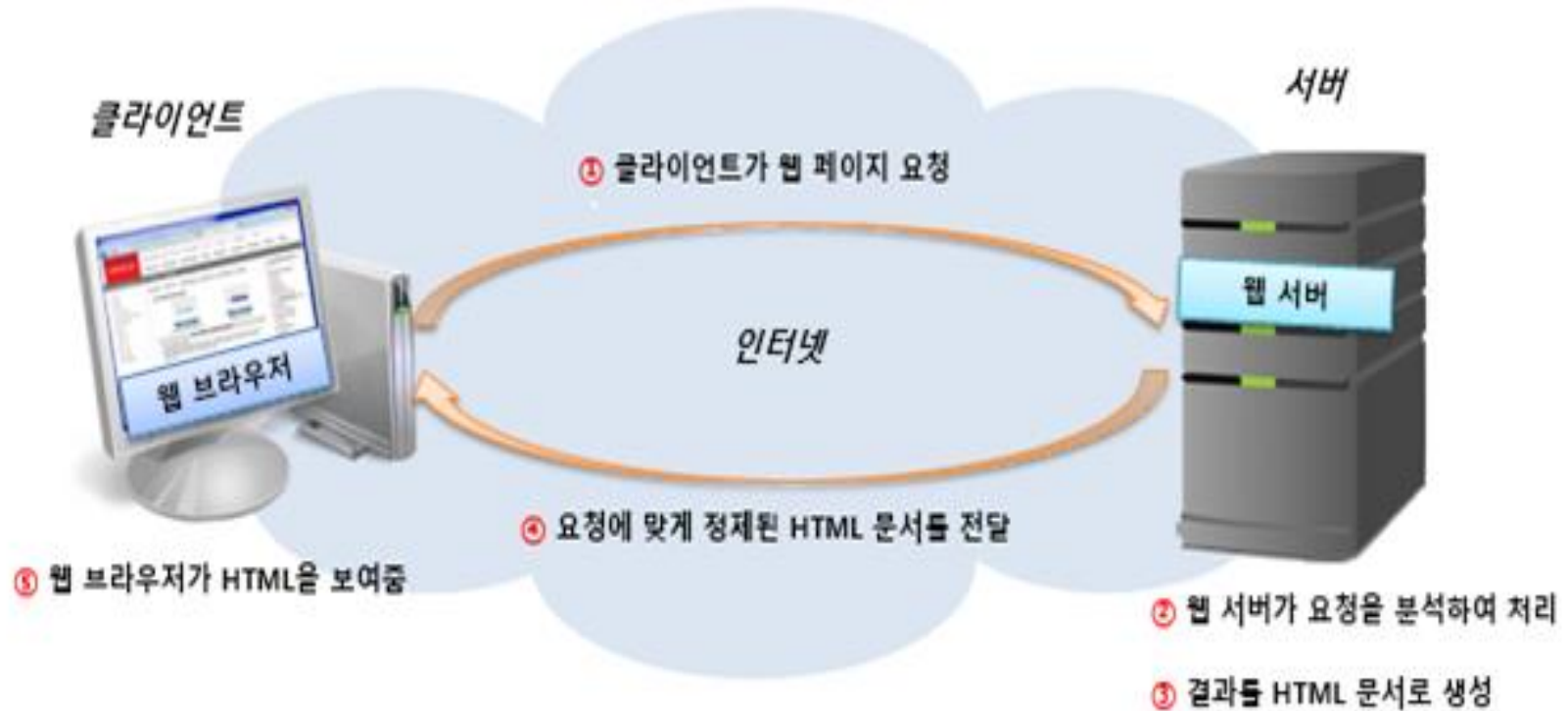
1. 웹과 JSP 프로그래밍 이해하기

❖ 정적 웹 페이지의 동작 방식



1. 웹과 JSP 프로그래밍 이해하기

❖ 동적 웹 페이지의 동작 방식



1. 웹과 JSP 프로그래밍 이해하기

❖ 웹 프로그래밍과 JSP

■ 웹 프로그래밍 언어

- 클라이언트 측 실행 언어와 서버 측 실행 언어로 구분
- 자바를 기반으로 하는 JSP는 서버 측 웹 프로그래밍 언어 중 하나

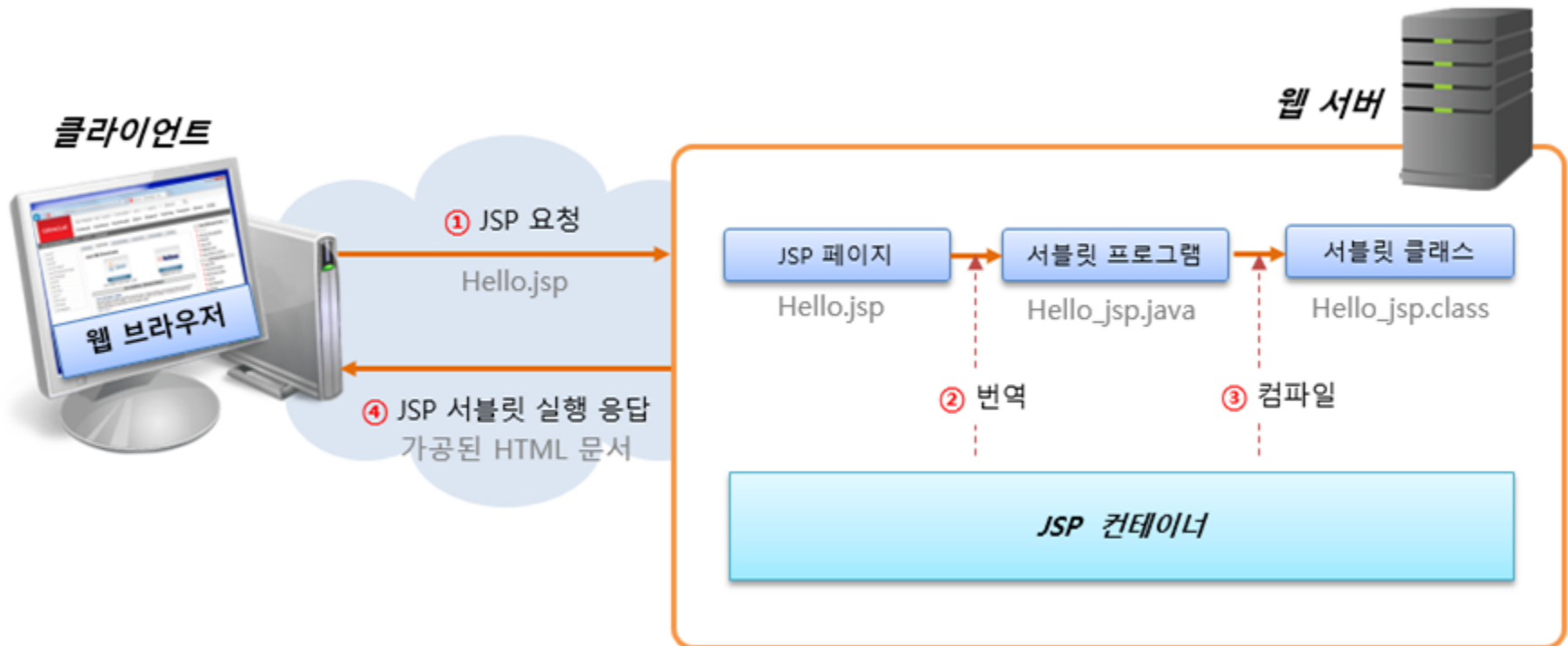
■ JSP의 특징

- JSP는 서블릿 기술의 확장입니다.
- JSP는 유지 관리가 용이합니다.
- JSP는 빠른 개발이 가능합니다.
- JSP로 개발하면 코드 길이를 줄일 수 있습니다.

1. 웹과 JSP 프로그래밍 이해하기

❖ JSP 페이지의 처리 과정(1)

- JSP 페이지는 하나의 서블릿 프로그램으로 변환되어 실행됨.
- JSP 컨테이너(JSP 엔진)의 역할
 - Hello.jsp 에서 Hello.jsp.java 라는 서블릿 프로그램을 생성함
 - Hello.jsp.java 서블릿 프로그램이 컴파일되어 Hello_jsp.class 를 생성함



1. 웹과 JSP 프로그래밍 이해하기

❖ JSP 페이지의 처리 과정(2)

- 웹 브라우저가 웹 서버에 JSP를 요청함
- 웹 서버는 요청된 Hello.jsp에서 .jsp 확장자를 발견하여 JSP 페이지임을 확인하고 웹 서버에 있는 JSP 컨테이너에 전달함.
- JSP 컨테이너는 JSP 페이지를 서블릿 프로그램인 Hello_jsp.java 로 변환함.
- JSP 컨테이너가 서블릿 프로그램을 컴파일하여 Hello_jsp.class 로 만들고 이를 웹 서버에 전달함.
- 웹 서버는 정적 웹 페이지처럼 *.class 파일의 실행 결과를 웹 브라우저에 응답으로 전달하므로 웹 브라우저는 새로 가공된 HTML 페이지를 동적으로 처리한 결과를 보여줌.

1. 웹과 JSP 프로그래밍 이해하기

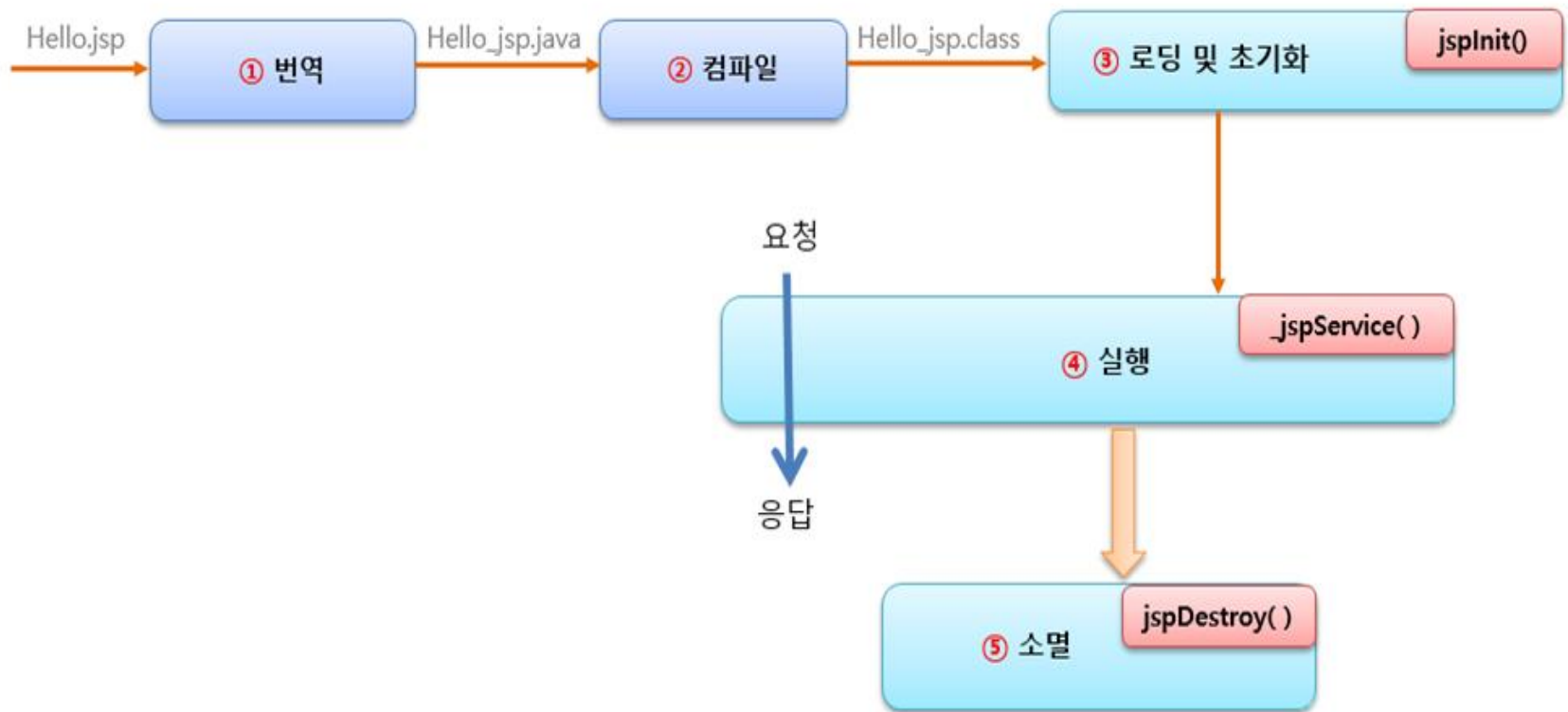
❖ JSP 페이지의 처리 과정(3)

- JSP 컨테이너(JSP 엔진)를 웹 서버와 분리하여 설치할 수도 있고, 웹 서버에 포함도 가능.
- Apache Tomcat 웹 서버는 JSP 컨테이너가 포함된 웹 서버임.

1. 웹과 JSP 프로그래밍 이해하기

❖ JSP 생명주기

- JSP 페이지를 컴파일한 *.class에는 `jspInit()`, `_jspService`, `jspDestroy()` 메서드가 존재하며, JSP 생성부터 소멸까지의 과정에서 다음과 같은 역할을 수행함.



1. 웹과 JSP 프로그래밍 이해하기

❖ JSP 생명주기

- ① 번역(translation) 단계
 - JSP 컨테이너가 JSP 소스 파일을 자바 코드(서블릿)으로 변환함
- ② 컴파일(compile) 단계
 - JSP 컨테이너가 번역 단계에서 생성된 자바 코드인 서블릿을 컴파일하여 클래스 파일을 생성
- ③ 로딩 및 초기화 단계
 - JSP 컨테이너가 앞의 두 단계에서 생성된 *.class 를 로딩하고 클래스의 인스턴스를 작성함.
- ④ 실행(execution) 단계
 - 각 클라이언트의 요청에 대해 JSP 컨테이너가 요청 및 응답 객체를 전달하는 `_jspService()` 메서드를 실행함.
- ⑤ 소멸(destruction) 단계
 - JSP 생명주기를 완료. JSP 컨테이너는 실행되고 있는 JSP를 `jspDestroy()` 메서드를 사용하여 제거함.

2. JSP 개발 환경 구축하기

❖ JSP 개발 환경 도구

- 자바 개발환경 : JDK
- 웹서버 : 톰캣
- 통합 개발 환경 :이클립스

요소	프로그램명	설명
자바 개발 환경	JDK	JSP는 HTML 코드 내에 자바 코드를 작성하기 때문에 자바 개발 도구인 JDK가 반드시 설치되어 있어야 합니다.
웹 서버	톰캣	웹 프로그래밍 언어로 작성된 웹 페이지가 실행되어 웹 브라우저에 나타나도록 하기 위해 웹 컨테이너를 설치해야 합니다. JSP 웹 컨테이너로 자주 사용되는 것 중에서 가장 유명한 것은 톰캣입니다. 톰캣은 오픈소스 프로젝트로서 무료로 누구나 사용할 수 있습니다.
통합 개발 환경	이클립스	JSP 코드를 작성한 후 이를 컴파일하여 오류를 검사하고 실행 결과를 확인할 수 있는 통합 개발 환경(IDE)으로 개발자에게 가장 인기 있는 이클립스(Eclipse)를 선택하여 설치합니다.

2. JSP 개발 환경 구축하기

❖ 자바 설치하고 환경 설정하기

- 자바 개발 키트
 - JDK(Java Development Kit)
- 자바 실행 환경
 - JRE(Java Runtime Environment)

JDK

컴파일러, 디버거, 애플릿뷰어(appletviewer) 등

JRE

클래스 로더(class loader), 자바 API, 실행시간 라이브러리 등

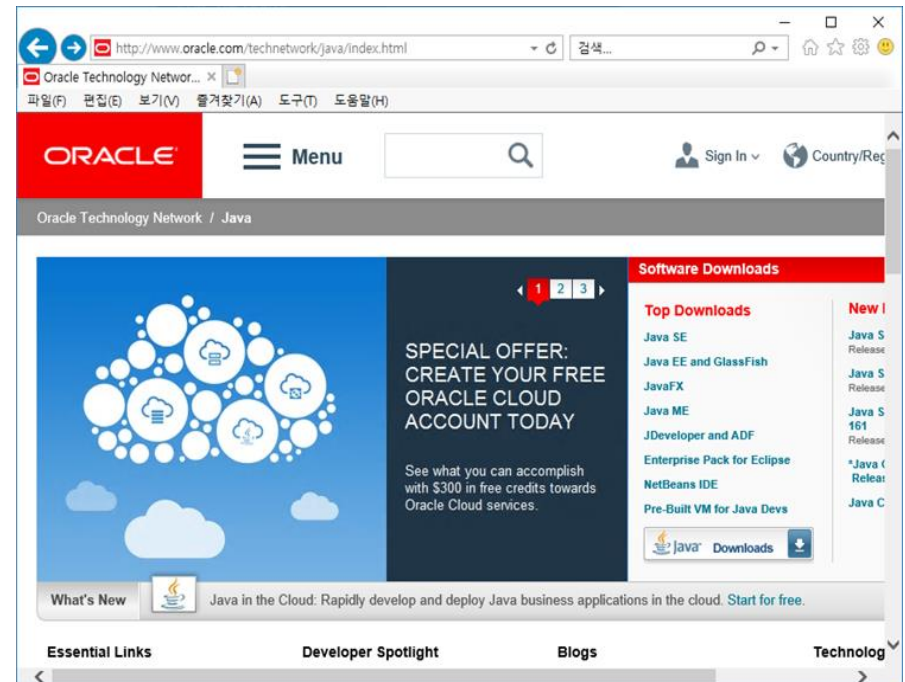
JVM

JIT(Java In Time Compiler), 자바 인터프리터 등

2. JSP 개발 환경 구축하기

예제 1-1 JDK 설치하기

- 1. 오라클 사이트에 접속하기
 - <http://www.oracle.com/kr/java>



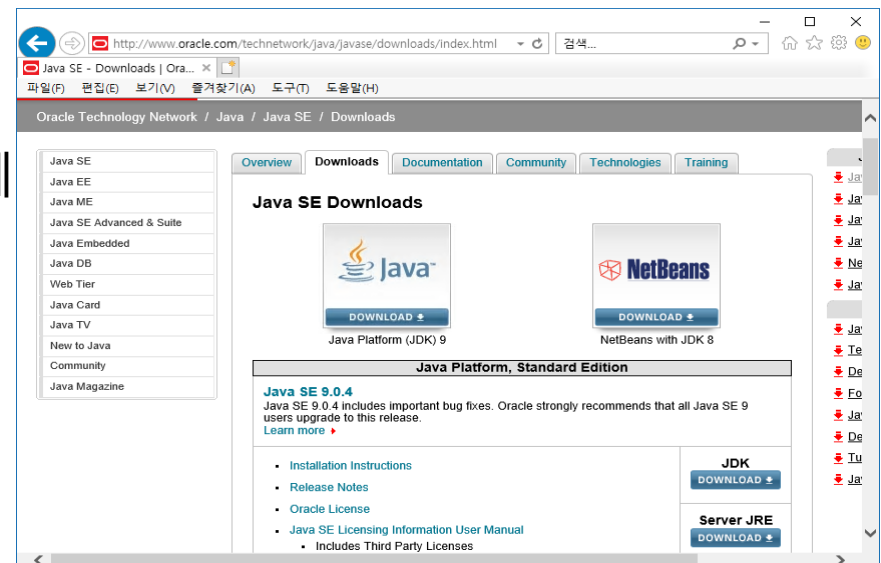
2. JSP 개발 환경 구축하기

예제 1-1 JDK 설치하기

■ 2.JDK 선택하기



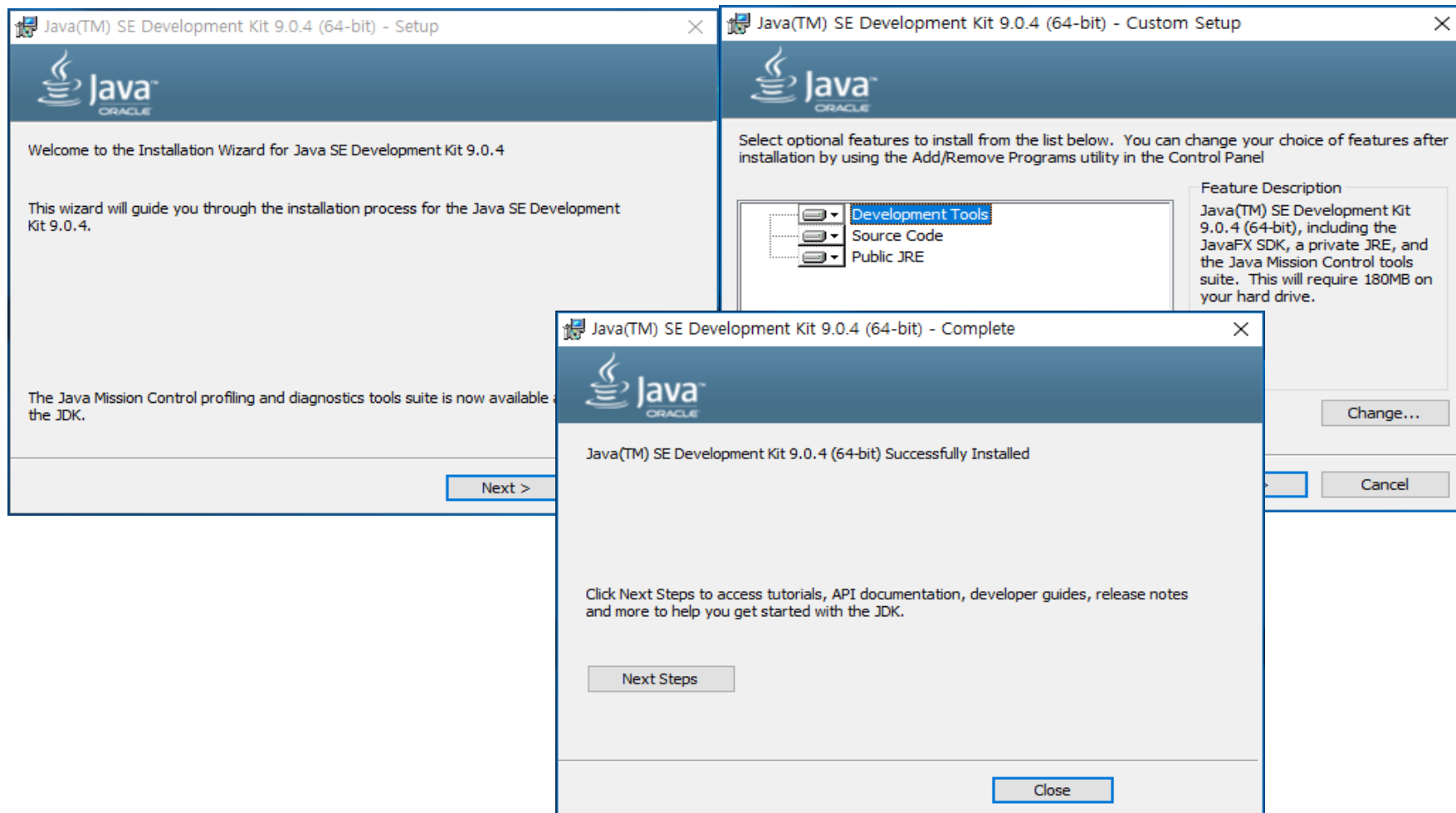
■ 3. 라이선스 동의 및 운영체제 버전에



2. JSP 개발 환경 구축하기

예제 1-1 JDK 설치하기

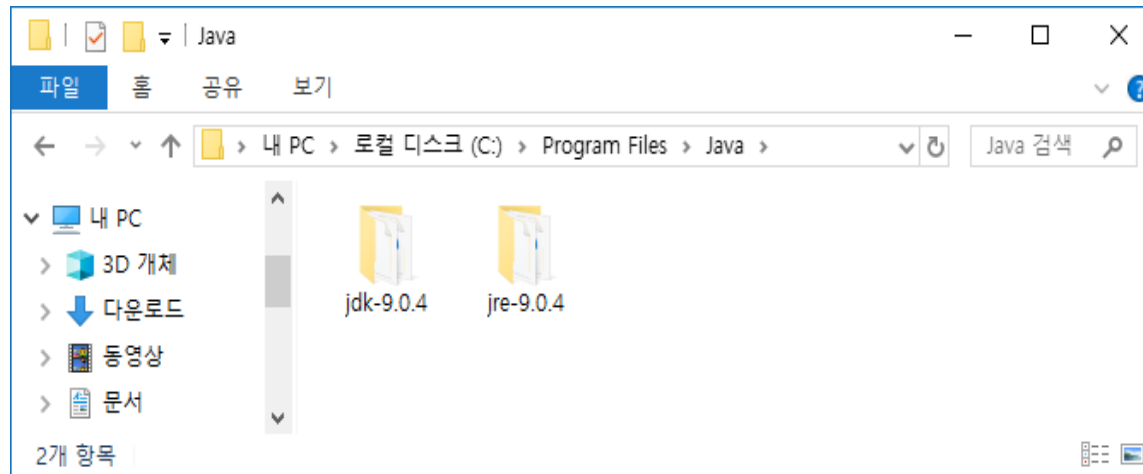
■ 4. 자바 설치하기



2. JSP 개발 환경 구축하기

예제 1-1 JDK 설치하기

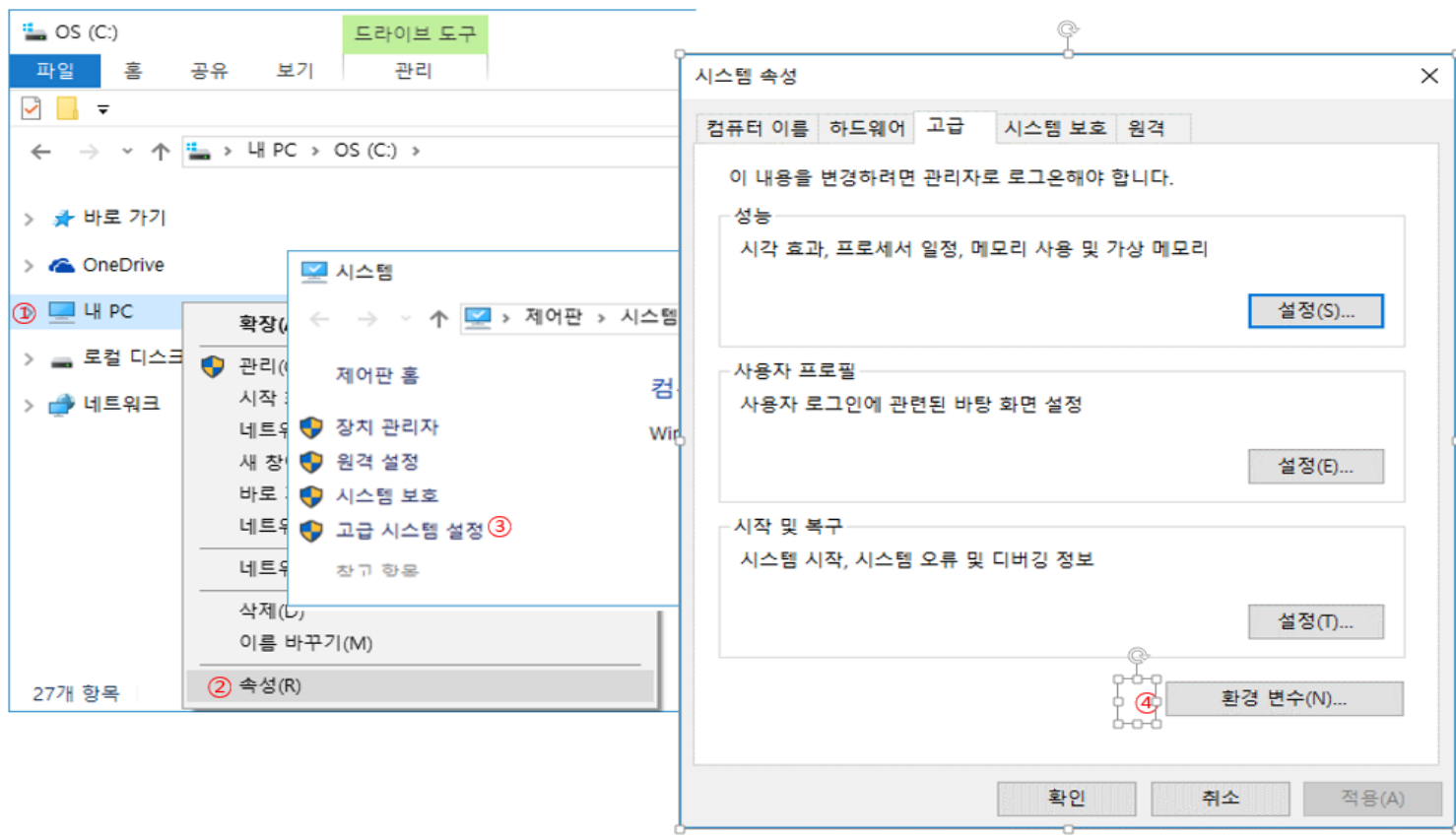
■ 5. 자바 설치 위치 확인하기



2. JSP 개발 환경 구축하기

예제 1-2 자바 환경 변수 설정하기

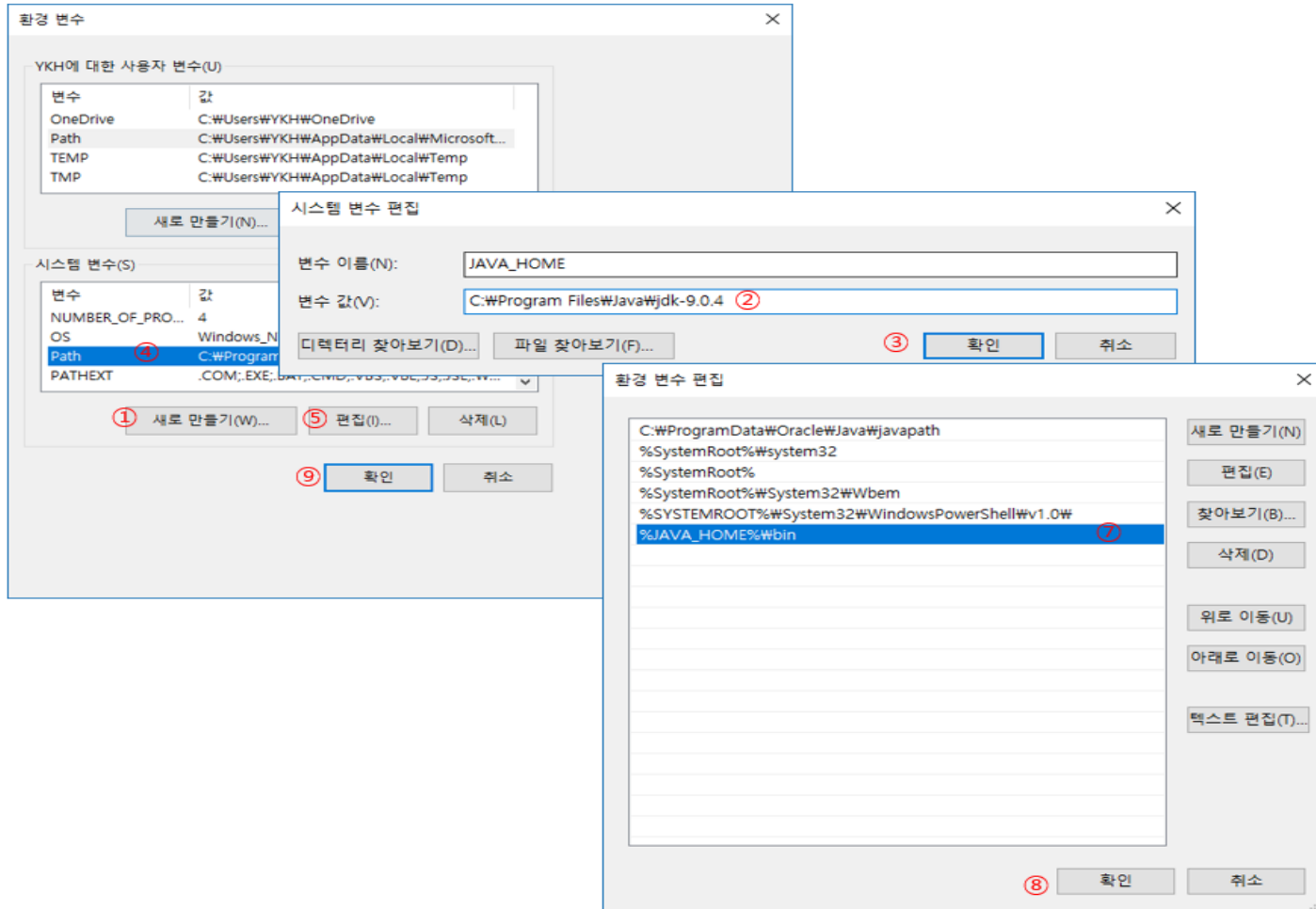
■ 1. [환경 변수] 대화상자 열기



2. JSP 개발 환경 구축하기

예제 1-2 자바 환경 변수 설정하기

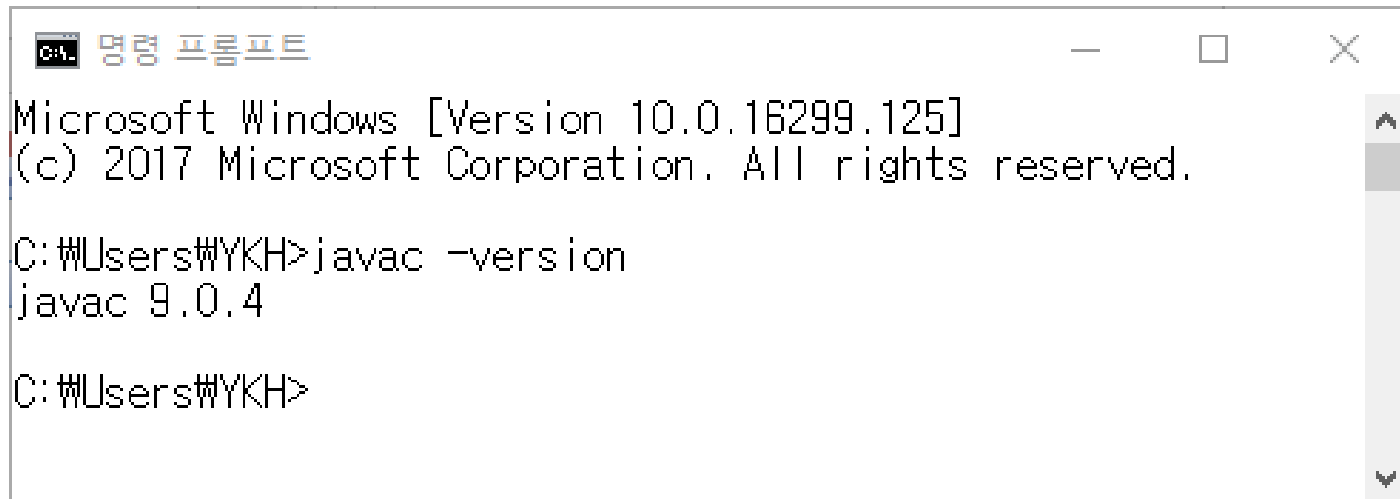
■ 2. 자바 환경 변수 설정하기



2. JSP 개발 환경 구축하기

예제 1-2 자바 환경 변수 설정하기

- 3. 자바에 설정한 환경 변수 확인하기



```
명령 프롬프트
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\₩YKH>javac -version
javac 9.0.4

C:\Users\₩YKH>
```

2. JSP 개발 환경 구축하기

❖ 웹 서버와 통합 개발 환경 설치하기

■ 웹 서버

- 톰캣

- 아파치 소프트웨어재단(Apache Software Foundation)에서 개발한 웹 애플리케이션 서버
- 자바 만들어진 웹 페이지를 구동하기 위한 엔진

■ 통합 개발 환경

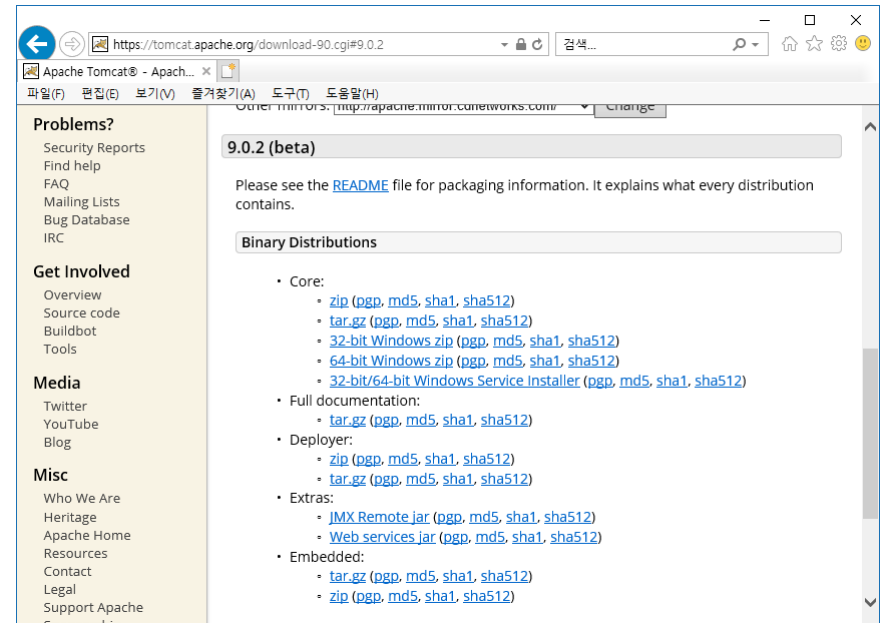
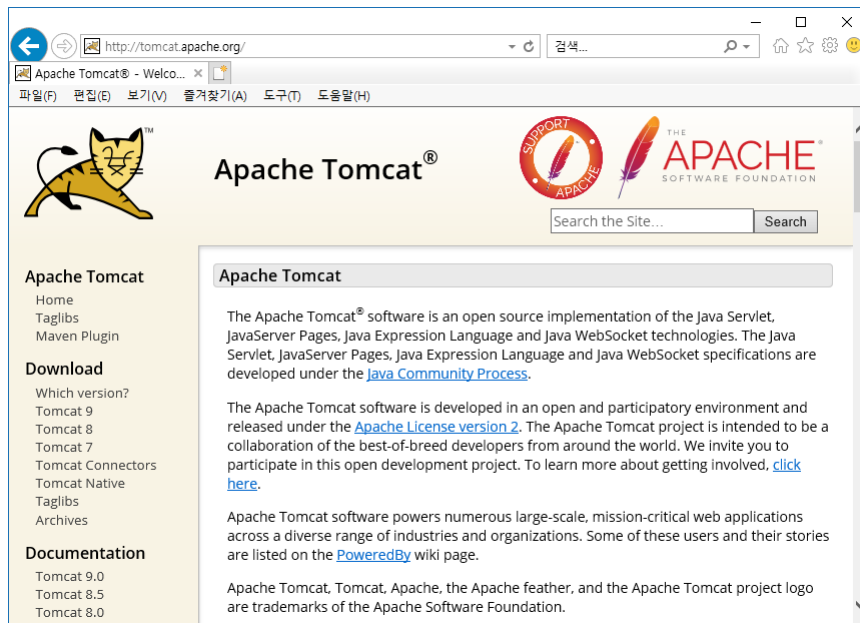
- 이클립스

- 자바 통합 개발 환경(IDE) 중 가장 많이 사용되는 개발 도구
- 자바를 기반으로 애플리케이션을 개발하기 위해 이클립스를 사용

2. JSP 개발 환경 구축하기

예제 1-3 톰캣 설치하기

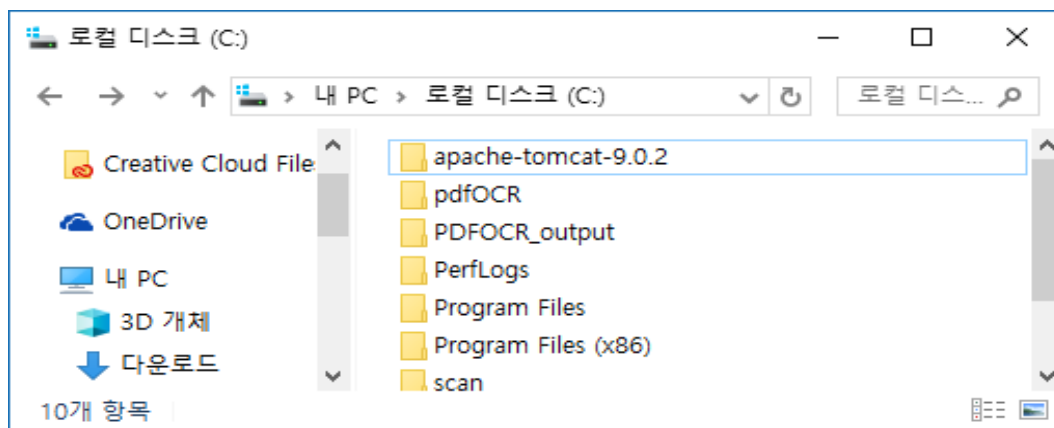
- 1. 아파치 사이트에 접속 및 다운로드하기
 - <http://tomcat.apache.org>



2. JSP 개발 환경 구축하기

예제 1-3 톰캣 설치하기

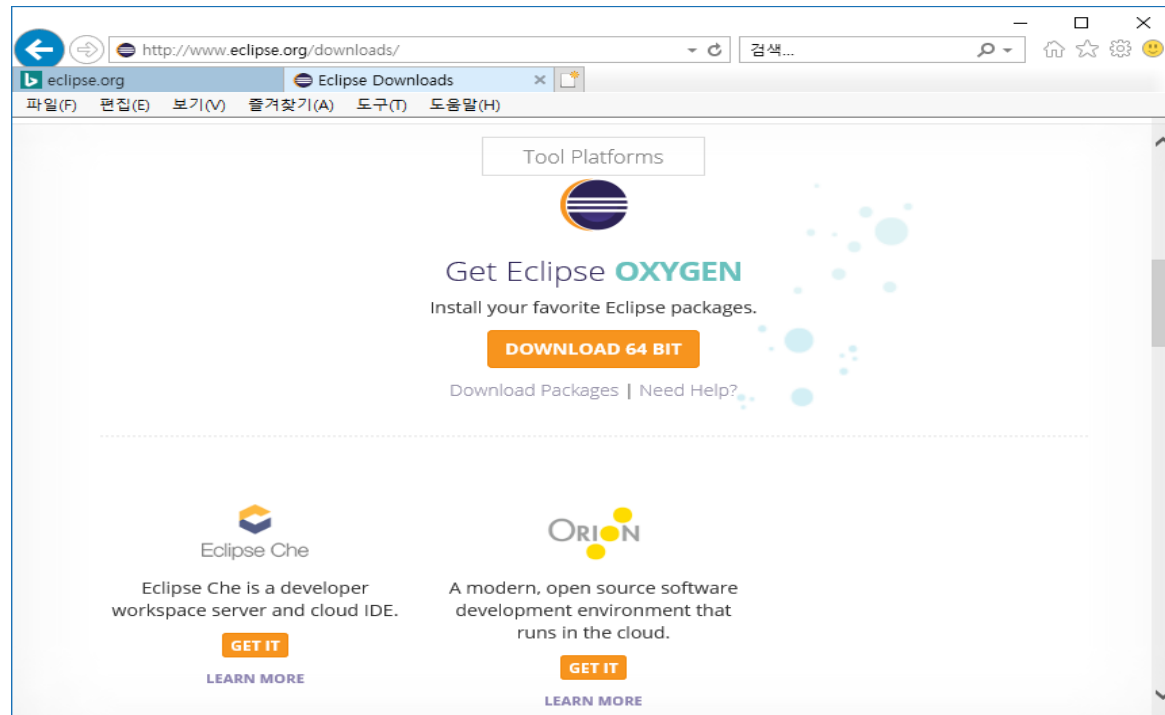
- 2. 다운로드한 apache-tomcat-9.0.2-windows-x64.zip의 압축을 풀고 하위에 있는 apache-tomcat-9.0.2 폴더를 C 드라이브로 옮겨줍니다.
- (C:\w 경로에 바로 압축을 풀어도 됩니다).



2. JSP 개발 환경 구축하기

예제 1-4 이클립스 다운로드하고 설치하기

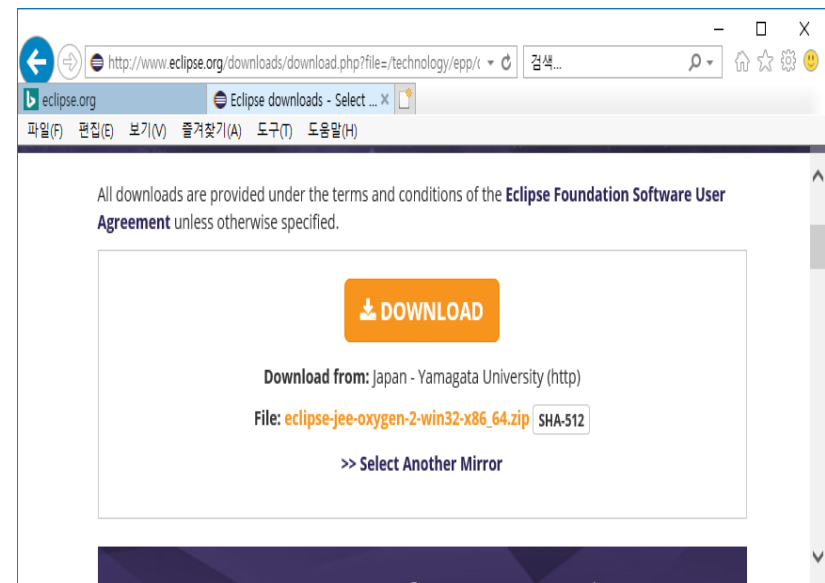
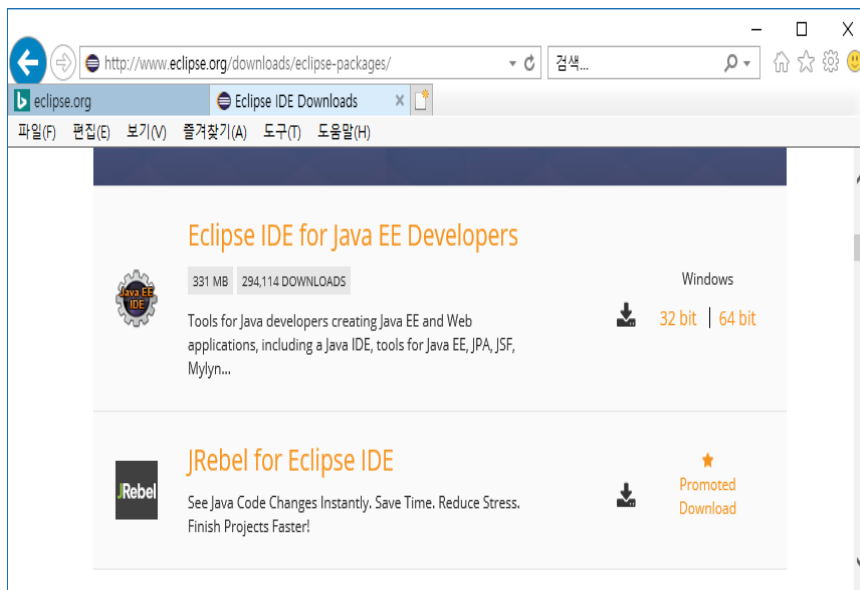
- 1. Eclipse 사이트에 접속하기
 - <http://www.eclipse.org/downloads/>



2. JSP 개발 환경 구축하기

예제 1-4 이클립스 다운로드하고 설치하기

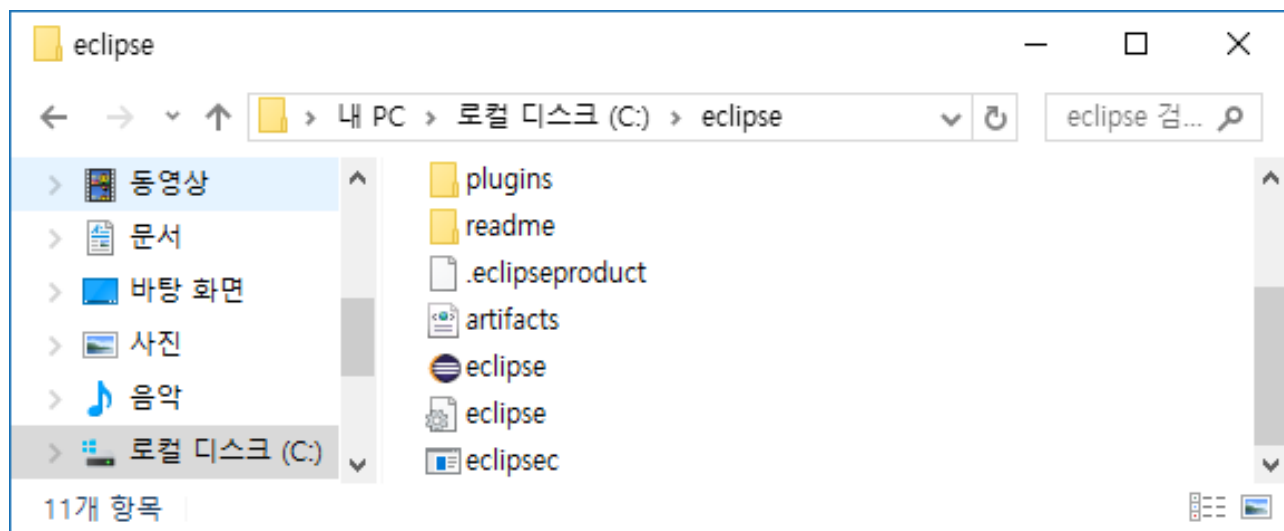
■ 2. 이클립스 설치 파일 다운로드하기



2. JSP 개발 환경 구축하기

예제 1-4 이클립스 다운로드하고 설치하기

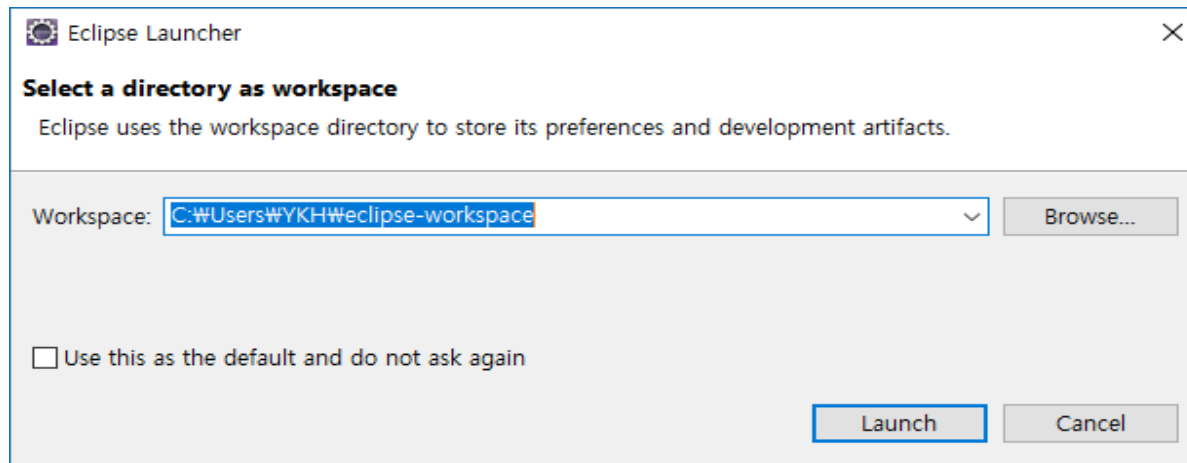
■ 3. 설치 완료하고 실행하기



2. JSP 개발 환경 구축하기

예제 1-4 이클립스 다운로드하고 설치하기

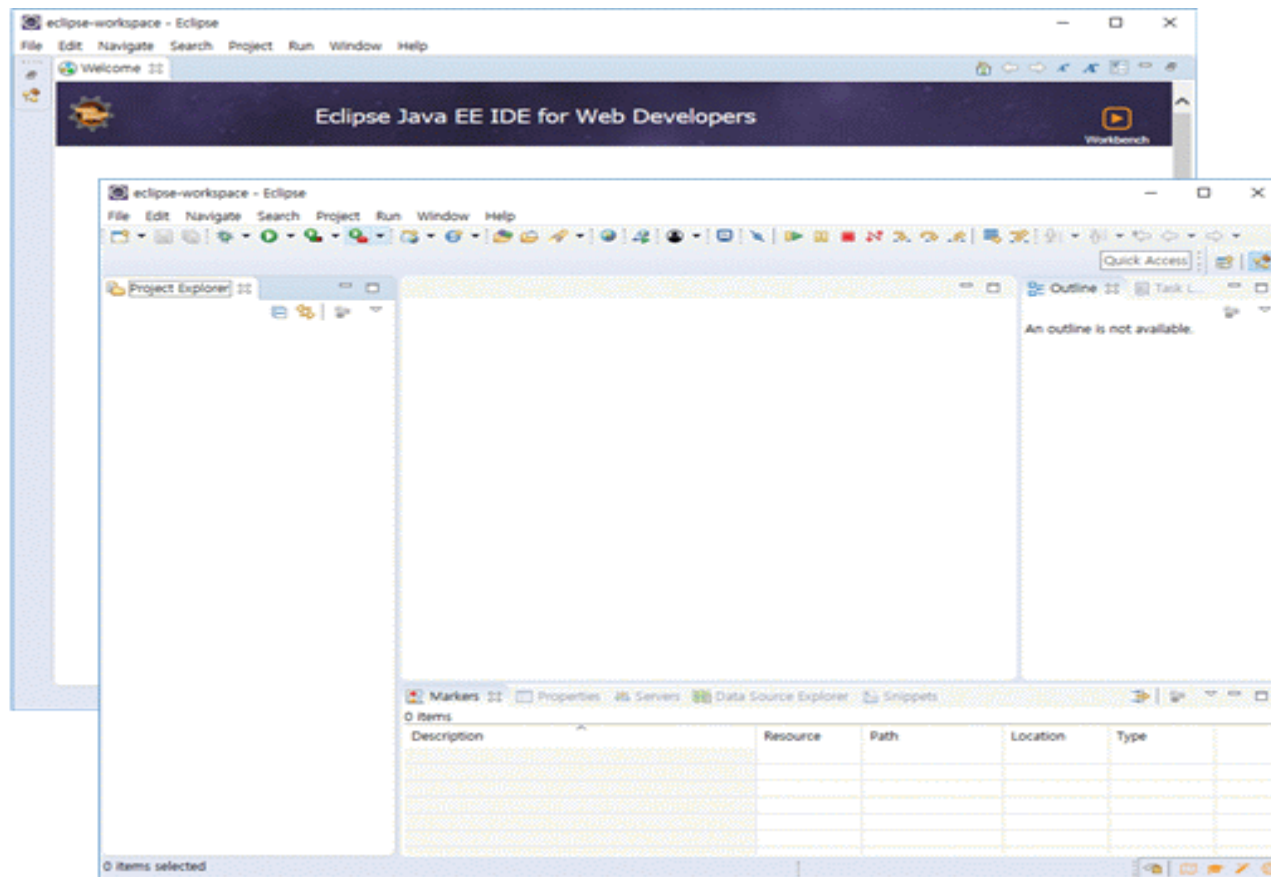
■ 4. 이클립스 작업 공간 설정하기



2. JSP 개발 환경 구축하기

예제 1-4 이클립스 다운로드하고 설치하기

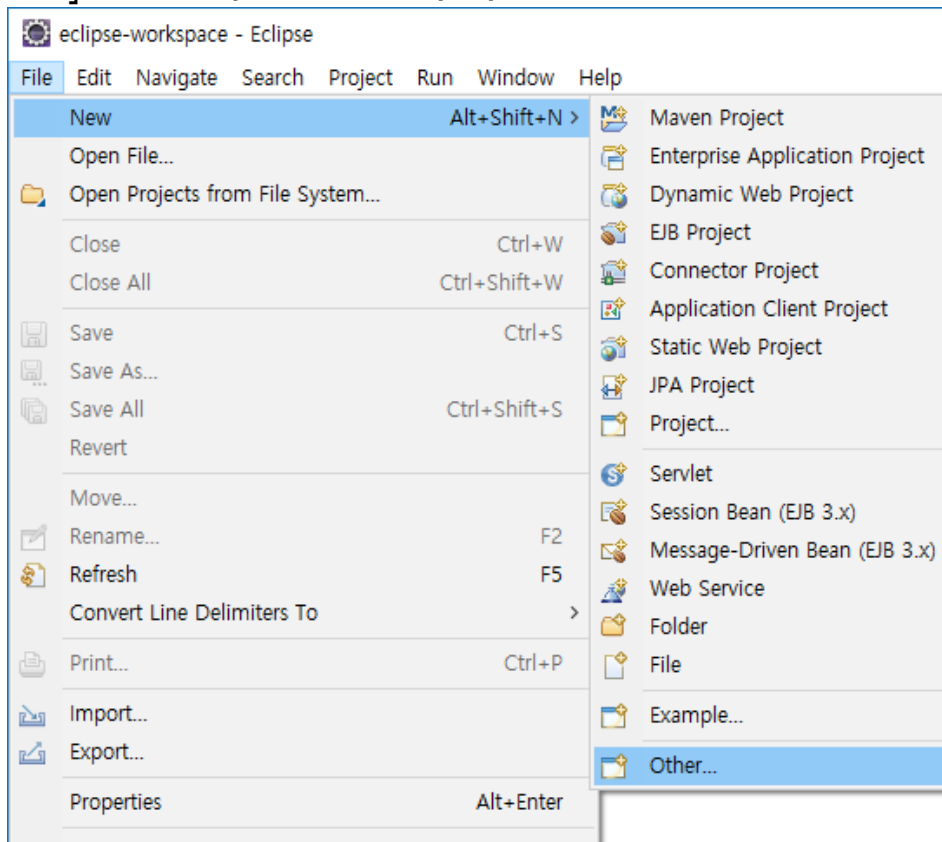
■ 5. 이클립스 실행 화면



2. JSP 개발 환경 구축하기

예제 1-5 이클립스와 톰캣 서버 연동하기

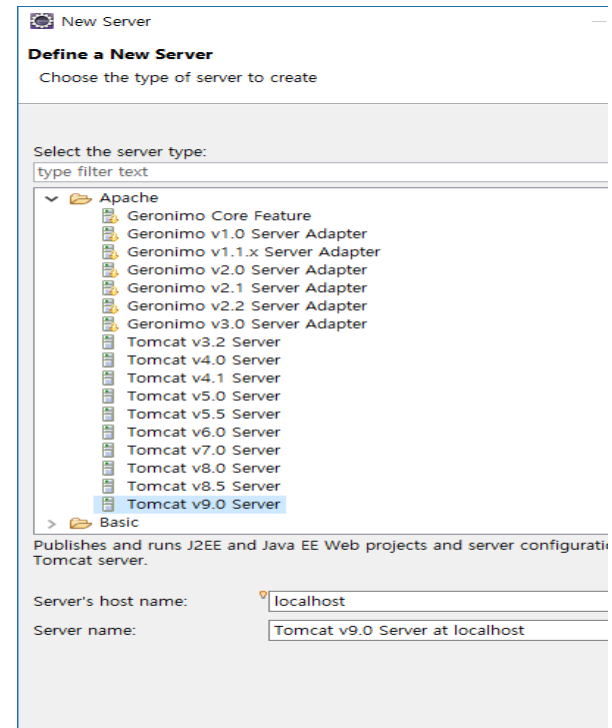
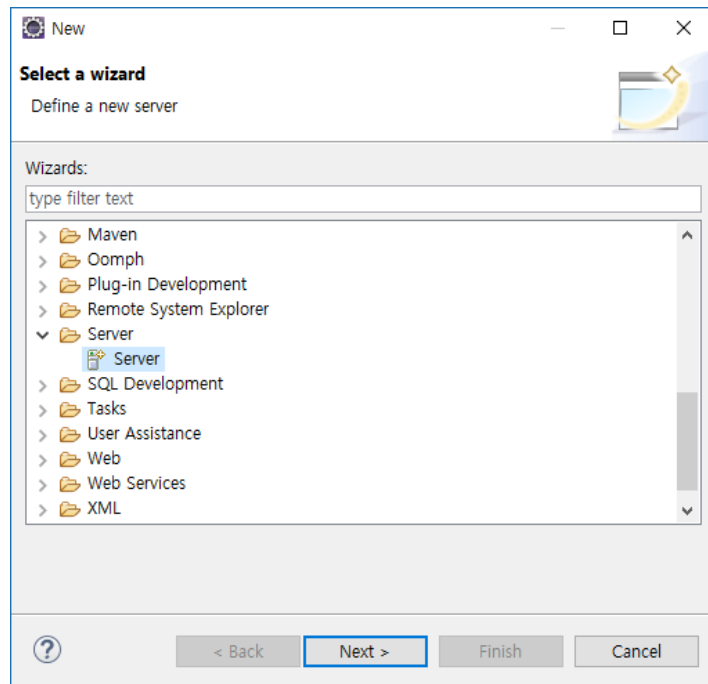
■ 1. [Server] 프로젝트 생성하기



2. JSP 개발 환경 구축하기

예제 1-5 이클립스와 톰캣 서버 연동하기

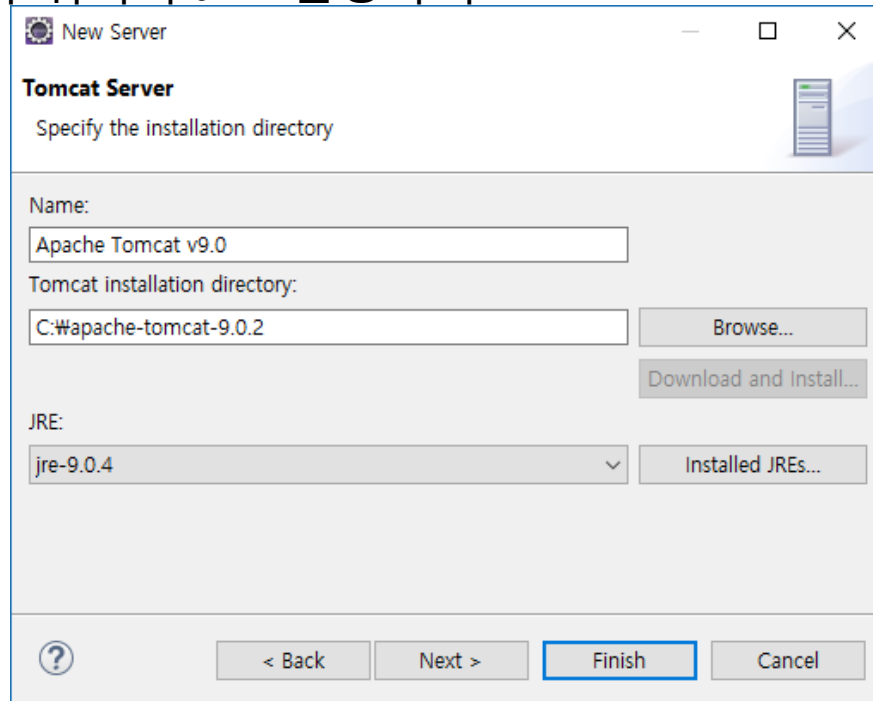
■ 2. 웹 서버 유형 설정하기



2. JSP 개발 환경 구축하기

예제 1-5 이클립스와 톰캣 서버 연동하기

■ 3. 웹 서버 위치와 JRE 설정하기



The image shows the 'New Server' dialog box in Eclipse IDE, specifically for configuring a 'Tomcat Server'. The dialog has a title bar with a question mark icon and standard window controls. The main content area is titled 'Tomcat Server' and includes a subtitle 'Specify the installation directory'. Below this, there are three main sections: 'Name:', 'Tomcat installation directory:', and 'JRE:'. The 'Name' field contains 'Apache Tomcat v9.0'. The 'Tomcat installation directory' field contains 'C:\wapache-tomcat-9.0.2', with a 'Browse...' button to its right. Below the directory field is a 'Download and Install...' button. The 'JRE' section has a dropdown menu showing 'jre-9.0.4' and an 'Installed JREs...' button to its right. At the bottom of the dialog, there are four buttons: a help button (question mark), '< Back', 'Next >', and 'Finish' (which is highlighted with a blue border), and a 'Cancel' button.

New Server

Tomcat Server
Specify the installation directory

Name:
Apache Tomcat v9.0

Tomcat installation directory:
C:\wapache-tomcat-9.0.2

JRE:
jre-9.0.4

2. JSP 개발 환경 구축하기

예제 1-5 이클립스와 톰캣 서버 연동하기

■ 4. 연동 확인하기

The screenshot displays the Eclipse IDE interface with the 'Servers' view on the left. Under the 'Servers' folder, 'Tomcat v9.0 Server at localhost-config' is expanded, showing configuration files like catalina.policy, catalina.properties, context.xml, server.xml, tomcat-users.xml, and web.xml. The 'Servers' view at the bottom shows 'Tomcat v9.0 Server at localhost' with a status of '[Stopped, Republish]'. The 'Preferences' dialog is open, showing the 'Server Runtime Environments' tab. The 'Server runtime environments' table lists 'Apache Tomcat v9.0' as the type.

Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

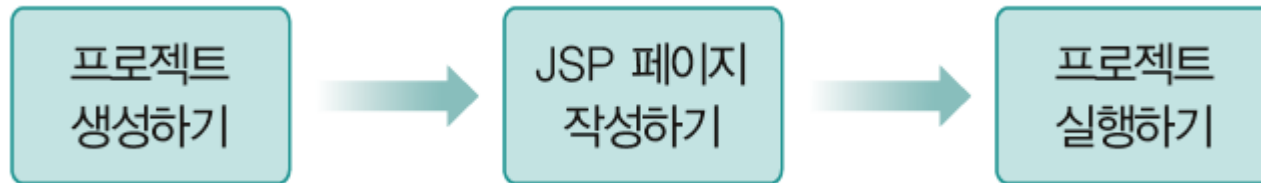
Name	Type
Apache Tomcat v9.0	Apache Tomcat v9.0

Buttons: Add..., Edit..., Remove, Search..., Columns...

Buttons: Apply and Close, Cancel

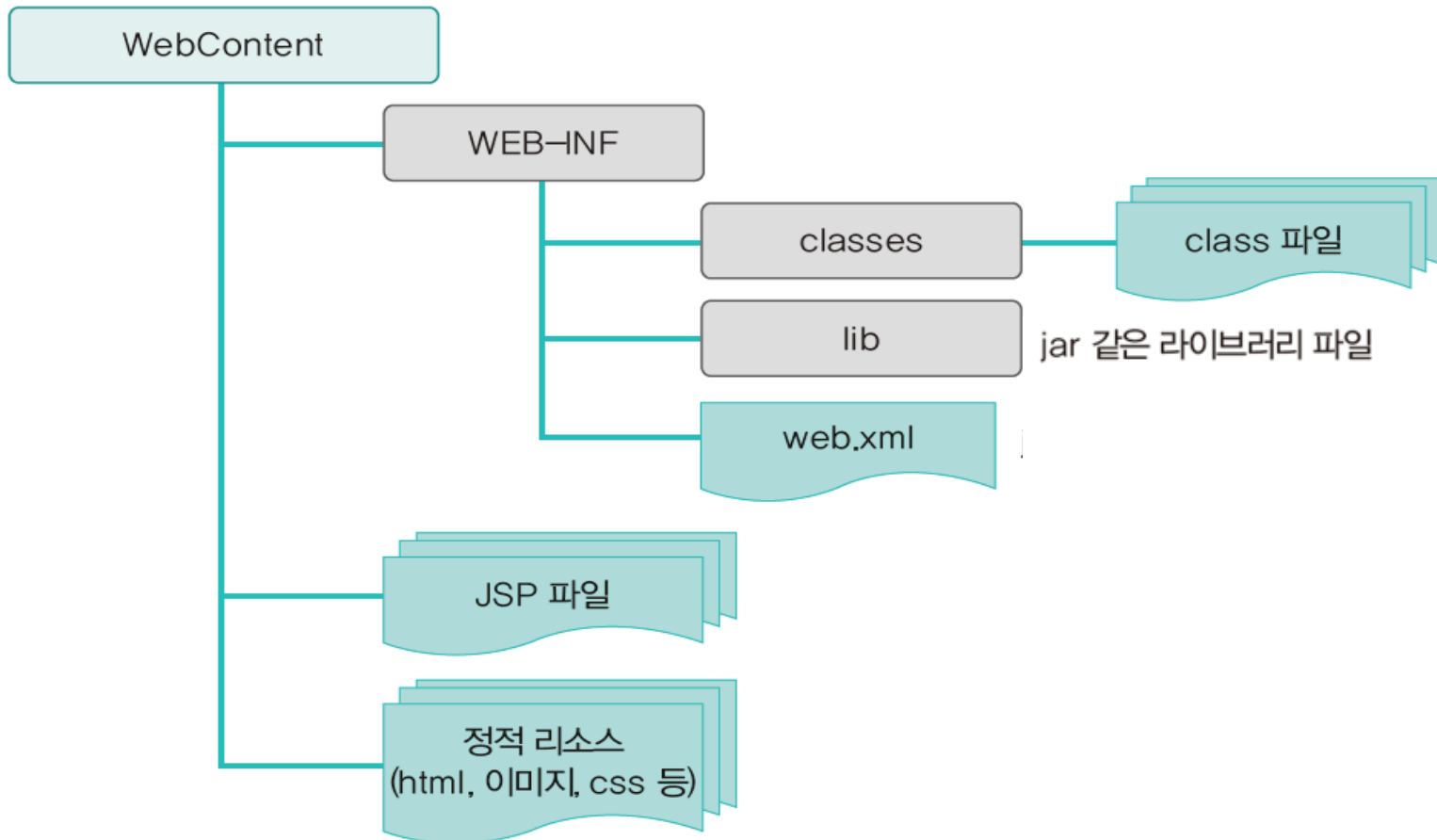
2. JSP 개발 환경 구축하기

❖ 프로젝트 만들고 실행하기



2. JSP 개발 환경 구축하기

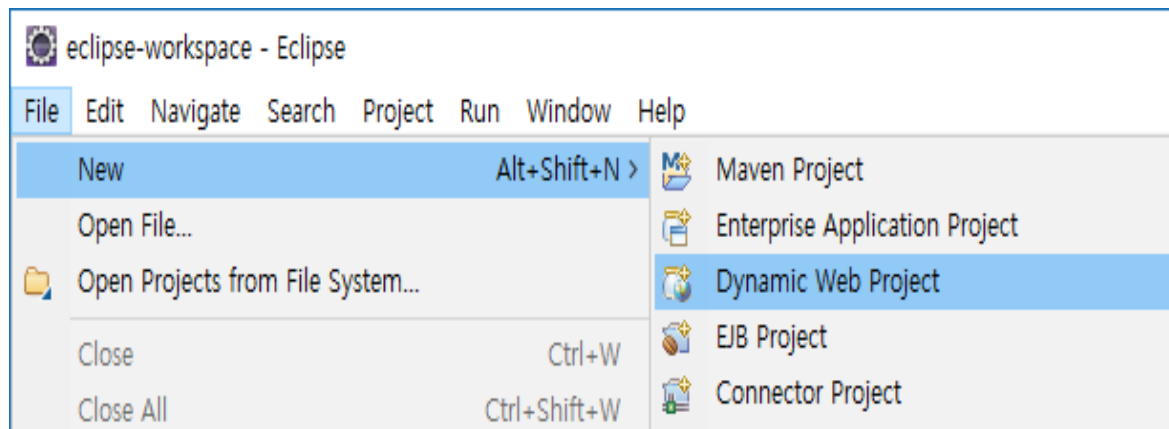
❖ 동적 웹 프로젝트의 구조



2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

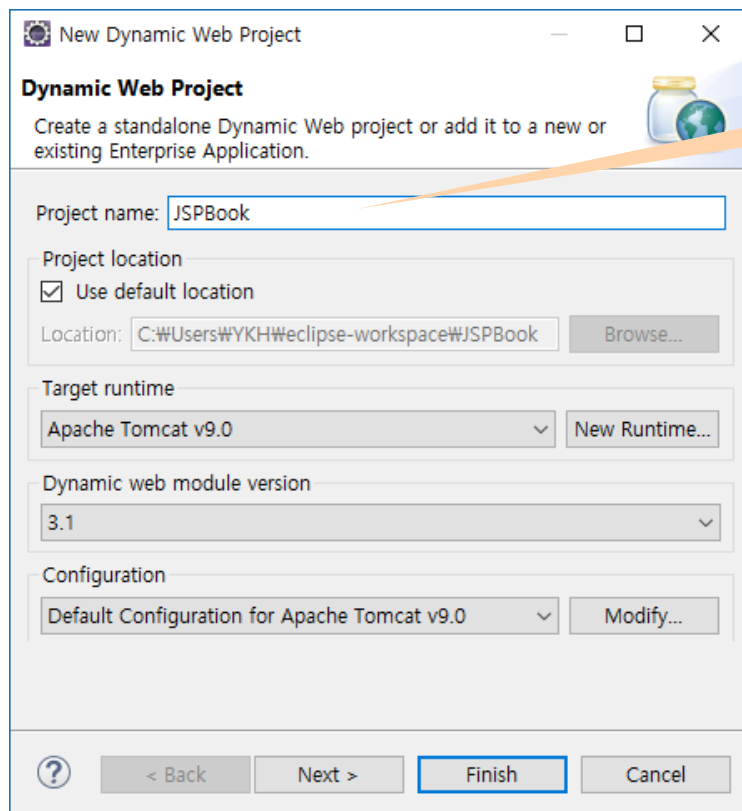
■ 1. 동적 웹 프로젝트 생성하기



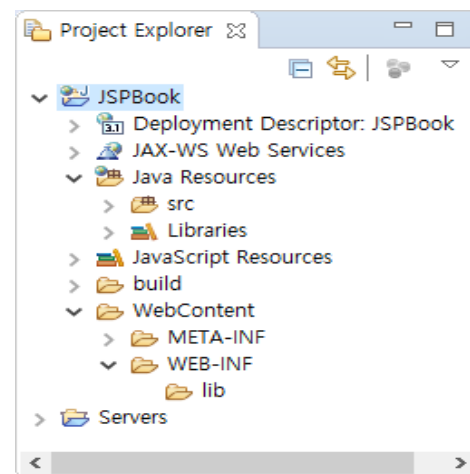
2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

■ 2. 프로젝트명 설정하기



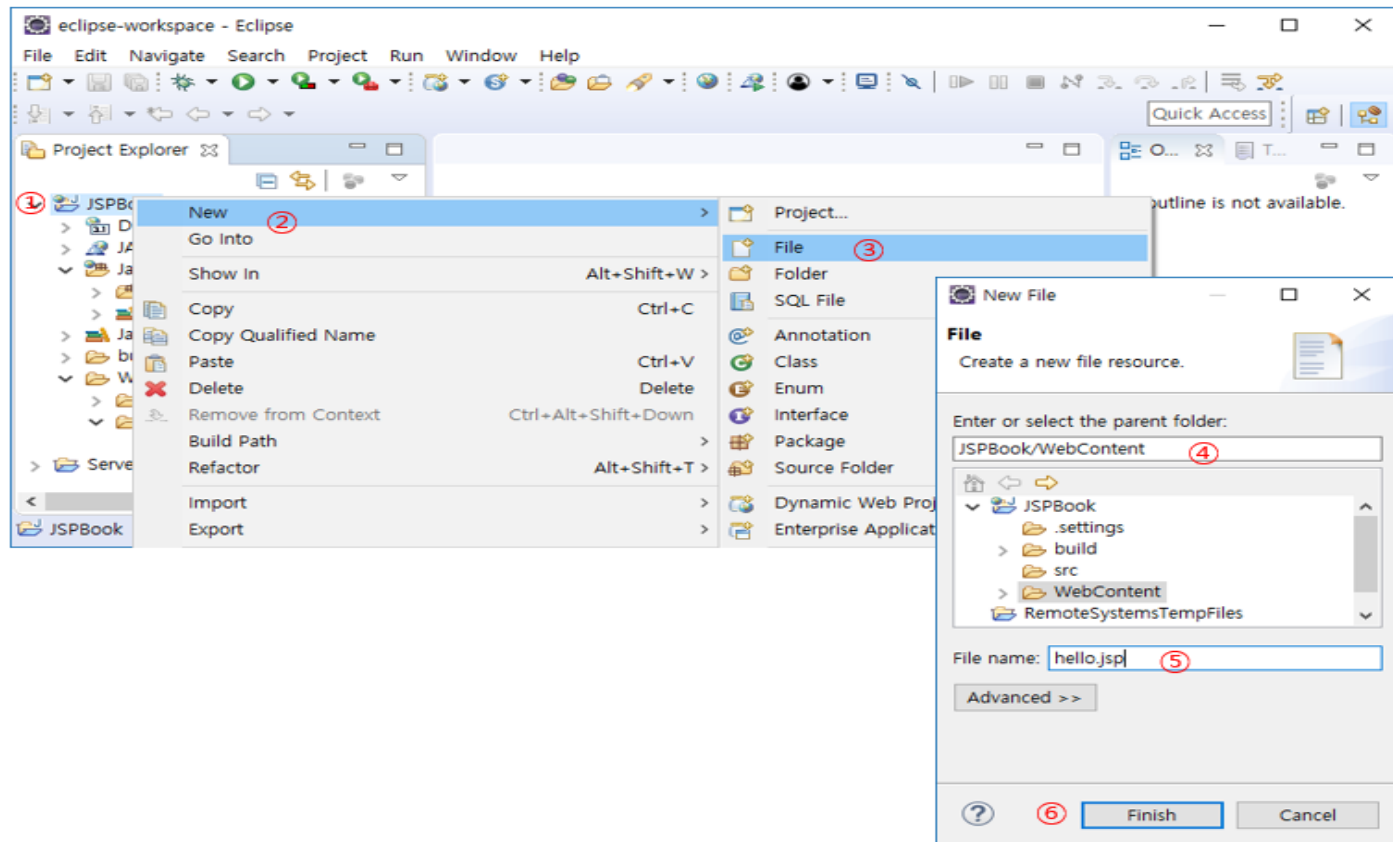
JSPBook 입력



2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

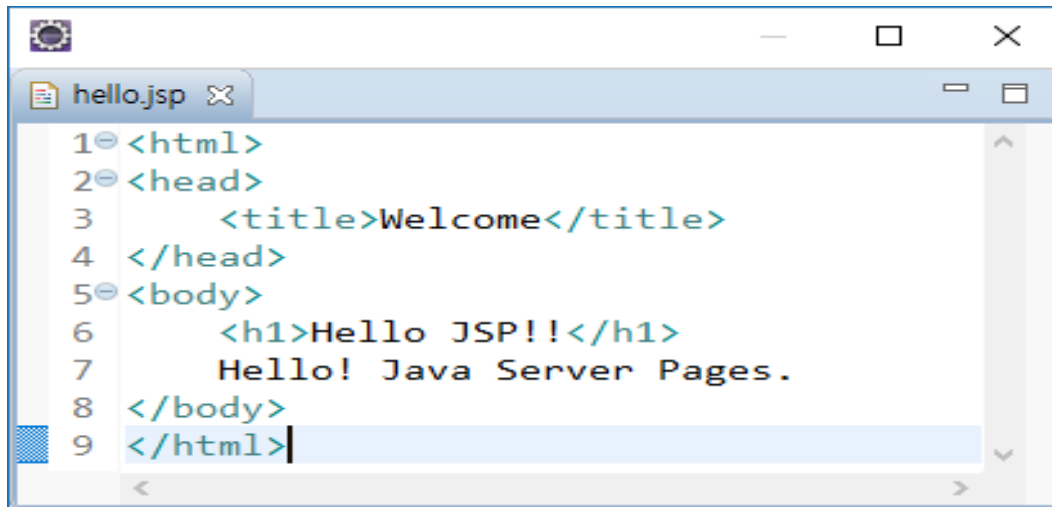
■ 3. JSP 파일 생성하기



2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

■ 4. JSP 페이지 코드 작성하기

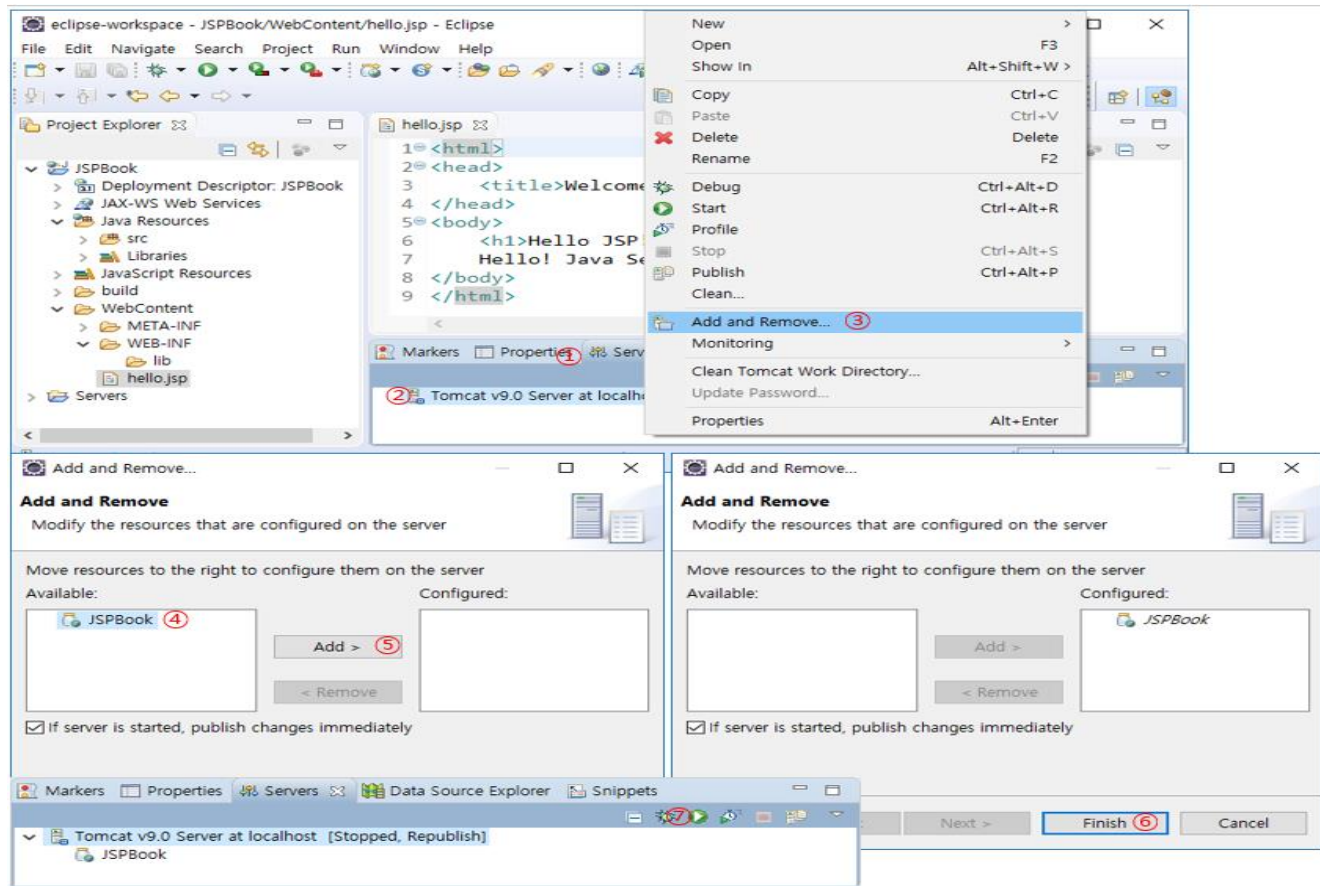


```
1 <html>
2 <head>
3     <title>Welcome</title>
4 </head>
5 <body>
6     <h1>Hello JSP!!</h1>
7     Hello! Java Server Pages.
8 </body>
9 </html>
```


2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

■ 5. 톰캣 서버에 등록하기



2. JSP 개발 환경 구축하기

예제 1-6 프로젝트 만들고 실행하기

■ 6. 실행 결과 확인하기

The screenshot displays the Eclipse IDE interface with the 'Run On Server' dialog open. The dialog prompts the user to select a server. Under the 'localhost' folder, 'Tomcat v9.0 Server at localhost' is listed with a 'Started' status. The 'Finish' button is highlighted.

The Eclipse editor shows the following JSP code:

```
1 <title>Welcome</title>
2 </head>
3 <body>
4   <h1>Hello JSP!!</h1>
5   Hello! Java Server Pages.
6 </body>
```

The 'Servers' tab at the bottom shows 'Tomcat v9.0 Server at localhost [Started, Synchronized]' and 'JSPBook [Synchronized]'.

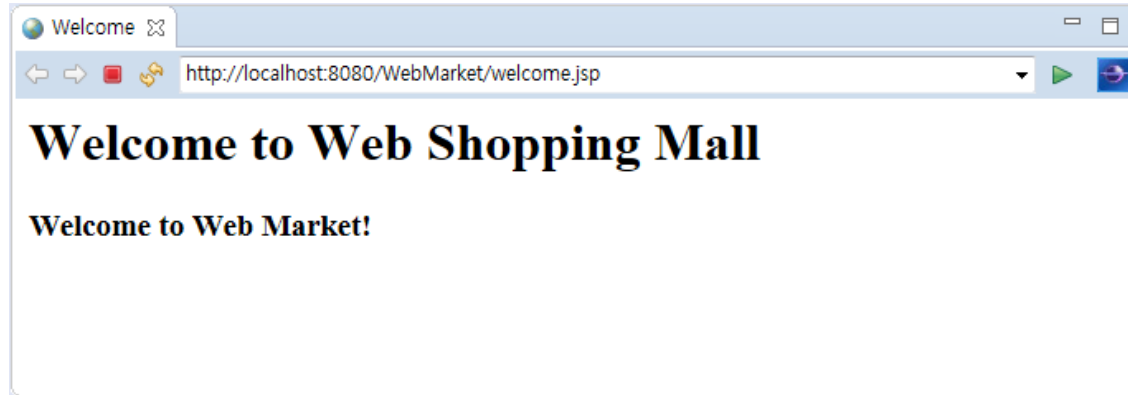
The browser window at the bottom shows the URL `http://localhost:8080/JSPBook/hello.jsp` and the rendered output:

Hello JSP!!

Hello! Java Server Pages.

3. (웹 쇼핑몰) 프로젝트 생성하기

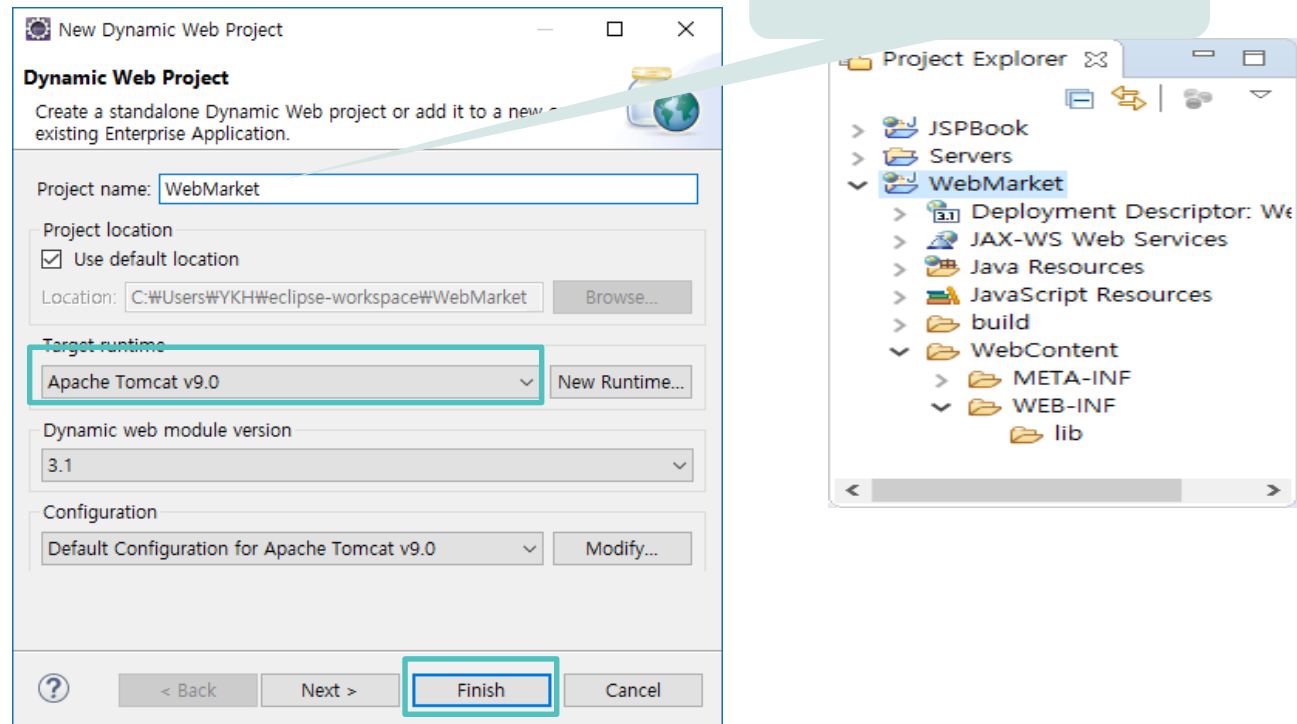
예제 1-7 웹 쇼핑몰 프로젝트 만들기



3. (웹 쇼핑몰) 프로젝트 생성하기

❖ [프로젝트 생성]

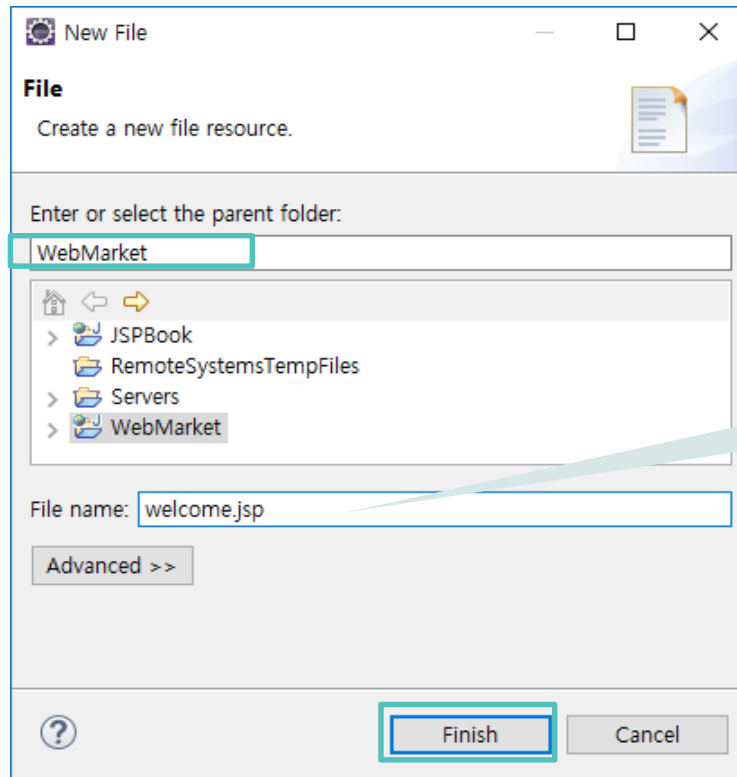
- 1. 동적 웹 프로젝트 생성하기
 - 이클립스의 [File]-[New]-[Dynamic Web Project]를 선택
- 2. 프로젝트명 설정하기



3. (웹 쇼핑몰) 프로젝트 생성하기

❖ [JSP 페이지 작성하기]

■ 3. JSP 파일 생성하기



welcome.jsp 입력

3. (웹 쇼핑몰) 프로젝트 생성하기

❖ [JSP 페이지 작성하기]

- 4. JSP 페이지 코드 작성하기

WebMarket/WebContent/welcome.jsp

```
01 <html>
02 <head>
03 <title>Welcome</title>
04 </head>
05 <body>
06     <h1>Welcome to Web Shopping Mall</h1>
07     <h3>Welcome to Web Market!</h3>
08 </body>
09 </html>
```

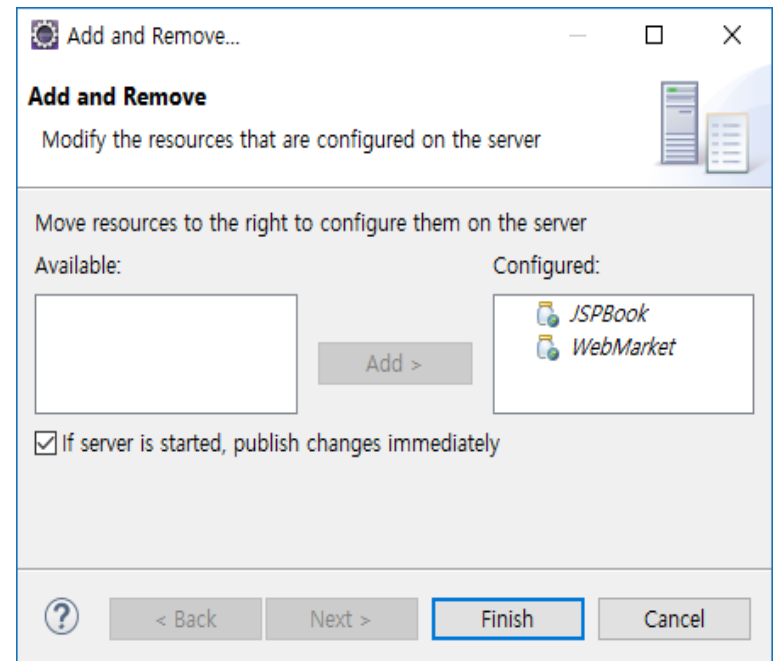
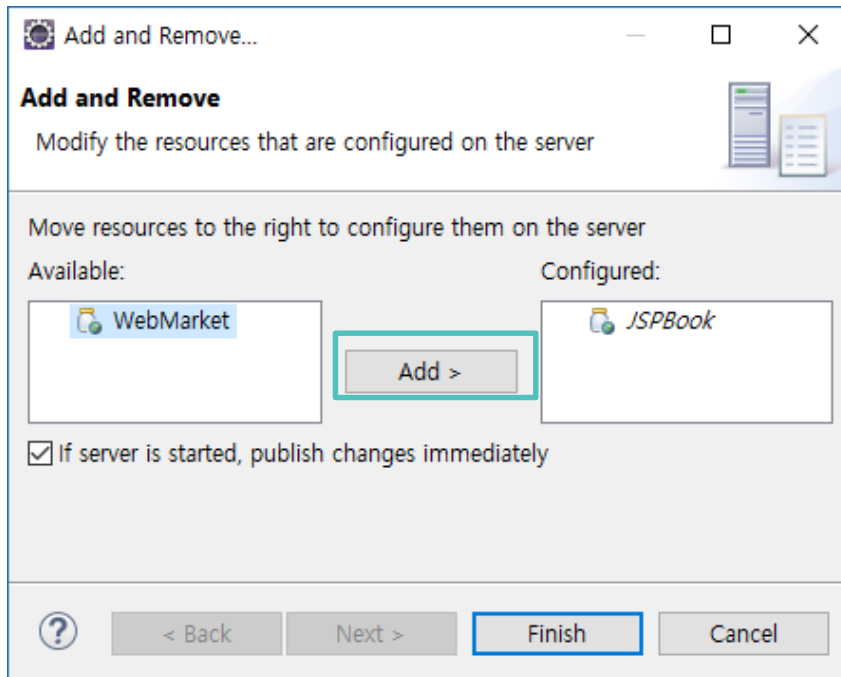
3. (웹 쇼핑몰) 프로젝트 생성하기

❖ [프로젝트 실행하기]

■ 5. 톰캣 서버에 등록하기

- 이클립스의 아래쪽 창 [Servers] 탭에 있는 Tomcat v9.0 Server at localhost에서 마우스 오른쪽 버튼을 누르고 [Add and Remove]를 선택

■



3. (웹 쇼핑몰) 프로젝트 생성하기

❖ [프로젝트 실행하기]

■ 6. 실행 결과 확인하기

