



JSP의 개발 환경 설정

이번 Chapter에서는 JSP 웹 애플리케이션을 개발하기 위한 환경을 설정해보겠다. JDK, 톰캣, 이클립스를 다운로드하여 설치한 후 가상 환경인 이클립스에서 개발하기 위한 환경 설정 방법을 학습하고, 가상 환경인 이클립스에서 개발한 웹 애플리케이션을 실제 서버 환경으로 배포하는 방법도 알아본다.



JDK 다운로드 및 설치

JDK 다운로드하기

JDK 설치하기

자바 환경 변수 설정하기

JDK 설치 확인하기



웹 컨테이너 톰캣 다운로드 및 설치

톰캣 다운로드하기

톰캣 설치하기



통합개발환경 이클립스 다운로드 및 설치

이클립스 다운로드 및 설치하기

이클립스 실행 및 작업 환경 설정하기



이클립스에서 웹 애플리케이션 작성

웹 서버 작성하기

동적 웹 프로젝트 작성하기

JSP 페이지와 서블릿 작성

웹 애플리케이션 배포 - WAR 내보내기



웹 애플리케이션 폴더 구조와 JSP의 처리 과정

웹 애플리케이션 폴더 구조

JSP의 처리 과정



JDK 다운로드 및 설치

- 자바 기반의 애플리케이션을 작성하려면 반드시 JDK(Java Development Kit, 자바 개발 키트)을 설치
- JDK는 자바 기반에서 작성되는 자바 프로그램밍 및 JSP 페이지를 실행할 수 있는 개발 환경 도구
 - 자바 언어를 컴퓨터가 인식할 수 있는 기계어로 번역해 프로그램을 실행



JDK 다운로드 및 설치

☺ JDK 다운로드

- ▣ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

☺ JDK 설치

- ▣ 다운로드 받은 JDK 설치 파일을 더블클릭해서 적당한 위치에 설치



JDK 다운로드 및 설치

자바 환경 변수 설정

환경 변수명	환경 변수 값
PATH	;C:\Program Files\Java\jdk1.8.0_05\bin;
CLASSPATH	.;C:\Program Files\Java\jdk1.8.0_05\lib\tools.jar
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_05



JDK 다운로드 및 설치

☺ JDK 설치 확인하기

- ▣ [명령 프롬프트] 창에서는 javac 명령어와 java 명령어를 실행
 - [보조 프로그램]-[명령 프롬프트] 메뉴를 사용해서 부름

웹 컨테이너 톰캣 다운로드 및 설치

- 웹 컨테이너인 톰캣(Tomcat)은 JSP 웹 애플리케이션을 작성할 수 있는 J2EE 영역을 제공
- 톰캣은 JSP 및 서블릿을 작성하고 실행할 수 있는 웹 컨테이너로 웹 서버와 웹 애플리케이션 서버의 기능도 가짐



웹 컨테이너 톰캣 다운로드 및 설치

☺ 톰캣 다운로드

▣ <http://tomcat.apache.org/>

☺ 톰캣 설치

▣ 다운로드 받은 톰캣 파일의 압축 해제

- 애플리케이션을 개발하기 위한 통합 개발 환경
- 체계적인 개발을 할 수 있는 각종 도구 제공
- 자바기반의 애플리케이션은 물론 C기반의 애플리케이션 및 PHP 기반의 웹 애플리케이션도 작성할 수 도 있으며 DBMS도 제어



통합 개발 환경 이클립스 다운로드 및 설치

이클립스 다운로드

▣ <http://www.eclipse.org/downloads/>

이클립스 설치

▣ 다운로드 받은 파일의 압축 해제



통합 개발 환경 이클립스 다운로드 및 설치

이클립스 실행

- eclipse.exe 파일 더블클릭
- 워크스페이스 지정

작업 환경 설정

- ▣ [Windows]-[Preference] 메뉴 선택
- ▣ 한글인코딩 지정
[Preference] 창의 [General]-[Workspace]
항목 선택
 - [Text file encoding]항목에서 [Other]를 선택 후
[UTF-8]을 선택

Ⓢ 웹 서버 작성

- 설치된 톰캣 서버를 이클립스에서 제어하도록 지정
- 서버 설정
이클립스에서 [File]-[New]-[Other] 메뉴 선택

동적 웹 프로젝트 작성

- 이클립스에서 애플리케이션은 프로젝트 단위로 프로그래밍
- 웹 애플리케이션은 동적 웹 프로젝트(Dynamic Web Project)로 생성해 실제 환경에 배포
- 동적 웹 프로젝트를 작성해야 웹 페이지를 작성

동적 웹 프로젝트 작성

- [File]-[New]-[Other] 메뉴 선택
[Web]-[Dynamic Web Project] 항목으로 생성
- 생성된 동적 웹 프로젝트는 서버에 추가

JSP 페이지와 서블릿 작성

▣ JSP 페이지 작성 및 실행

- JSP 페이지의 인코딩과 템플릿을 변경하고, JSP 페이지를 작성한 후 실행
 - 인코딩 : utf-8
- JSP 페이지는 [프로젝트]-[WebContent] 폴더에 생성
- [Run As]-[Run on Server] 메뉴를 사용해서 JSP 페이지를 실행

JSP 페이지와 서블릿 작성

- 서블릿(Servlet) 작성 및 실행
 - 서블릿은 [프로젝트]-[Java Ret] 폴더에 생성
 - [Run As]-[Run on Server] 메뉴를 사용해서 JSP 페이지를 실행

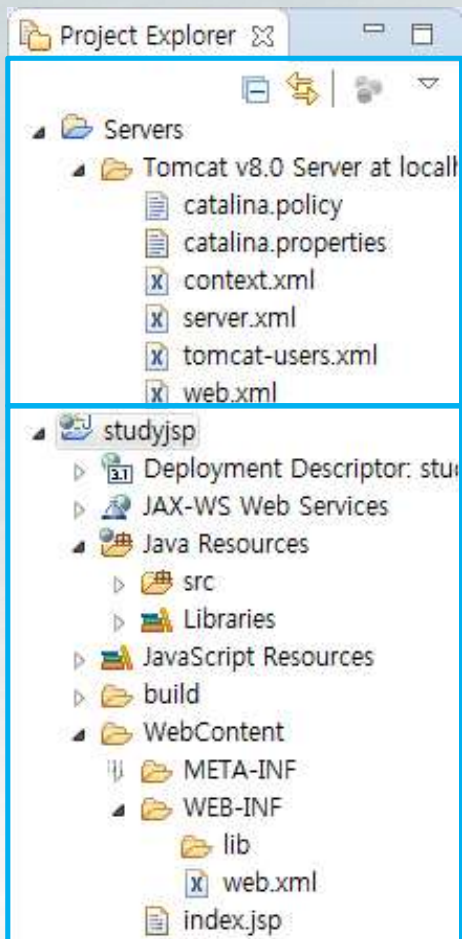
⦿ 웹 애플리케이션 배포 - WAR 보내기

- 웹 어플리케이션을 웹상에서 서비스를 하려면, 어플리케이션개발 후 실제로 서비스하는 서버로 배포파일인 WAR파일을 배포
- [Export]-[WAR file] 메뉴를 사용해서 배포

웹 애플리케이션 폴더 구조

- ▣ [Java Resources]-[src]
java 파일인 서블릿, 자바 로직 클래스를 작성하는 위치
- ▣ [WebContent]
JSP 페이지, HTML, CSS, JS 파일 및 이미지 파일 등이 위치

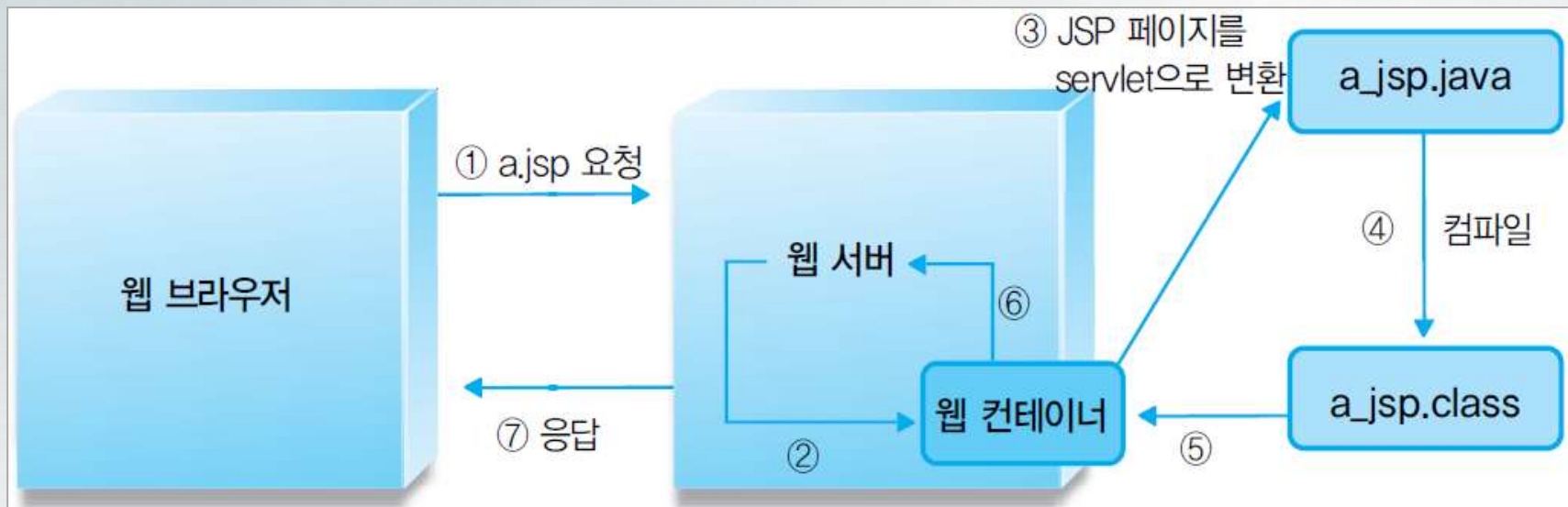
웹 애플리케이션 폴더 구조



이클립스의 가상 환경
웹 서버의 설정 파일

이클립스의 웹 애플리케이션이
되는 동적 웹 프로젝트

JSP의 처리 과정



▲ 단계별 JSP 페이지의 내부적 처리 과정

JSP의 처리 과정

- ① 사용자의 웹 브라우저에서 `http://서버주소/a.jsp` 와 같은 형태로 JSP 페이지를 요청
- ② 웹 서버는 요청한 해당 페이지를 JSP 컨테이너에 처리를 넘김
- ③ 해당 JSP 파일이 처음 요청된 것이라면 JSP 파일을 서블릿으로 파싱(`a.jsp` 페이지를 `a_jsp.java`로 변환)
-> 이전에 요청되었던 페이지일 경우 ⑥의 단계로 넘어감

JSP의 처리 과정

- ④ 클래스 파일로 컴파일(a_jsp.java 서블릿 파일은 a_jsp.class 파일로 컴파일)
- ⑤ 클래스 파일은 메모리에 로딩되어 실행
- ⑥ 실행결과는 다시 웹 서버에게 넘겨짐
- ⑦ 웹 서버는 웹 브라우저가 인식할 수 있는 HTML 형태로 결과를 웹 브라우저에게 응답