

JSP(Java Server page)

- I. JSP의 기본 개요
- II. JSP의 개발 환경설정

1. JSP의 기본 개요

- **웹 프로그래밍의 이해**
 - ▣ 웹 프로그래밍의 개요
 - ▣ 웹 프로그래밍 언어의 종류
 - ▣ 웹 애플리케이션의 구조 및 구성 요소
- **웹 애플리케이션 처리 방식 및 구현 방식**
 - ▣ 웹 애플리케이션 처리 방식
 - ▣ 웹 애플리케이션 구현 방식
- **서블릿의 개요**
 - ▣ 서블릿의 개요
 - ▣ JSP 개요

웹 프로그래밍의 이해

□ 웹 프로그래밍의 개요

- ▣ HTML만으로는 데이터가 실시간으로 변화하는 것을 처리하거나 저장하기에는 불가능
- ▣ 동적으로 변화하는 데이터를 처리하고 표시하기 위해서 개발된 것이 CGI, ASP, PHP, JSP
- ▣ 웹 프로그래밍은 기본적으로 클라이언트(Client)/서버(Server) 방식

웹 프로그래밍의 이해

□ 웹 프로그래밍의 개요

- 클라이언트(웹 브라우저)가 특정 페이지를 웹 서버에 요청(Request)-> 웹 서버가 처리 후->클라이언트(웹 브라우저)에게 응답(Response)



▲ 클라이언트/서버 방식의 구조

웹 프로그래밍의 이해

□ 웹 프로그래밍 언어의 종류

▣ CGI(Common Gateway Interface)

- 웹 서버와 동적 콘텐츠 생성을 맡은 프로그램 사이에서 정보를 주고받는 인터페이스
 - 개발 언어: UNIX 플랫폼(Platform)-펄(Perl), Windows 플랫폼(Platform)-비주얼 베이직(Visual Basic)
- 서버의 자원을 과도하게 사용하는 문제점 때문에 UNIX 플랫폼 이외에는 거의 사용되지 않음

▣ ASP(Active Server Page)

- 스크립트 방식으로 동적인 웹 페이지를 작성할 수 있도록 지원하는 기술
- ActiveX라는 제공된 컴포넌트를 사용할 수도 있으며, 이것을 직접 개발하기 위한 기능도 제공
- 특정 플랫폼과 특정 웹 서버에서만 동작
 - 윈도우즈 플랫폼에서 웹 서버로 IIS(Internet Information Server)만을 사용

웹 프로그래밍의 이해

□ 웹 프로그래밍 언어의 종류

▣ PHP(Personal HomePage tools, Professional Hypertext Preprocessor)

- C언어 기반, 서버에서 실행되는 스크립트 언어
- 배우기 쉽고, 개발 속도 빠름
- 컴포넌트 사용 불가, 보안에 취약한 단점
- 플랫폼에 독립적

▣ Servlet(Server + Applet)과 JSP(Java Server Pages)

- 자바 언어 기반으로 멀티스레딩(Multi Thread)에 의해 사용자 요구를 처리
- Servlet - 자바 코드에 의존적, JSP - 비교적 덜 의존적
- JSP와 Servlet은 같은 처리 구조를 가짐

웹 프로그래밍의 이해

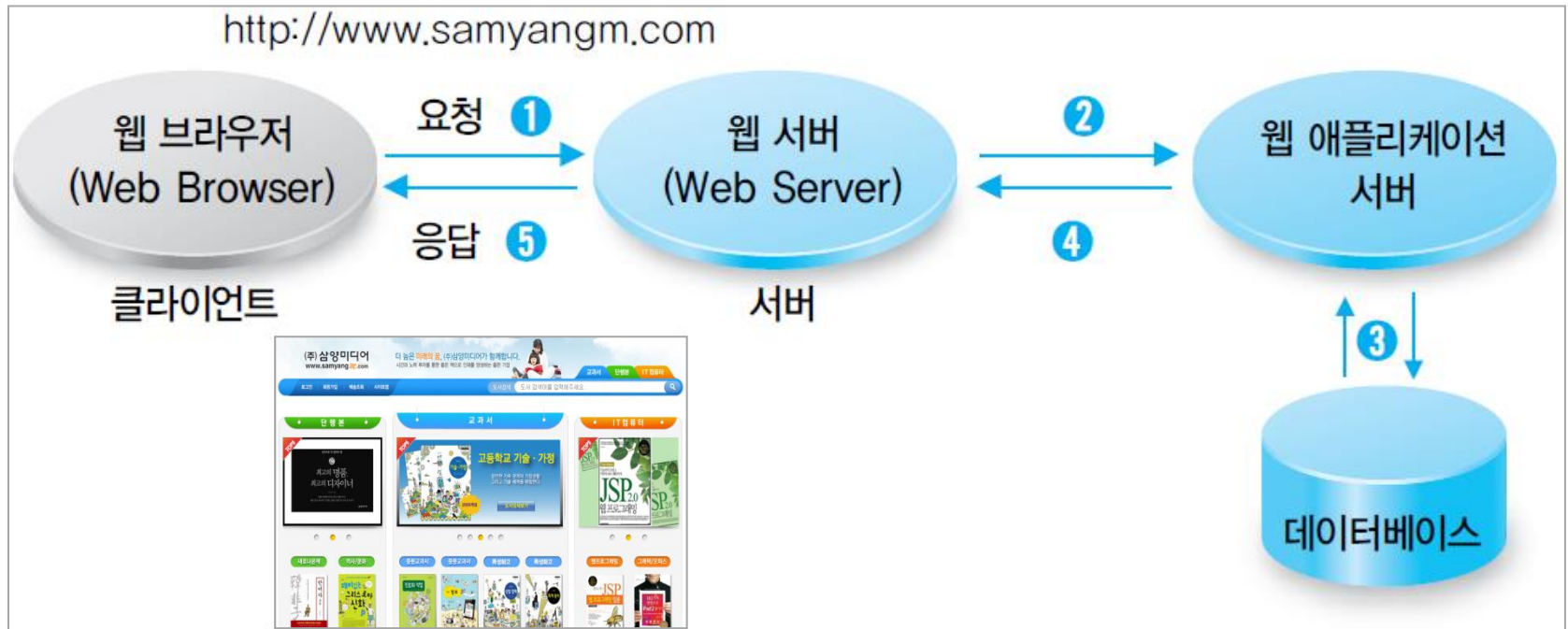
□ 웹 애플리케이션의 구조 및 구성 요소

▣ 웹 애플리케이션

- 웹을 기반으로 실행되는 프로그램을 의미
- 웹 프로그래밍을 통해 구현

웹 프로그래밍의 이해

- 웹 애플리케이션의 구조 및 구성 요소
 - ▣ 웹 애플리케이션의 구조



▲ 웹 애플리케이션의 구조 및 처리 순서

웹 프로그래밍의 이해

□ 웹 애플리케이션의 구조 및 구성 요소

▣ 웹 애플리케이션의 구성 요소

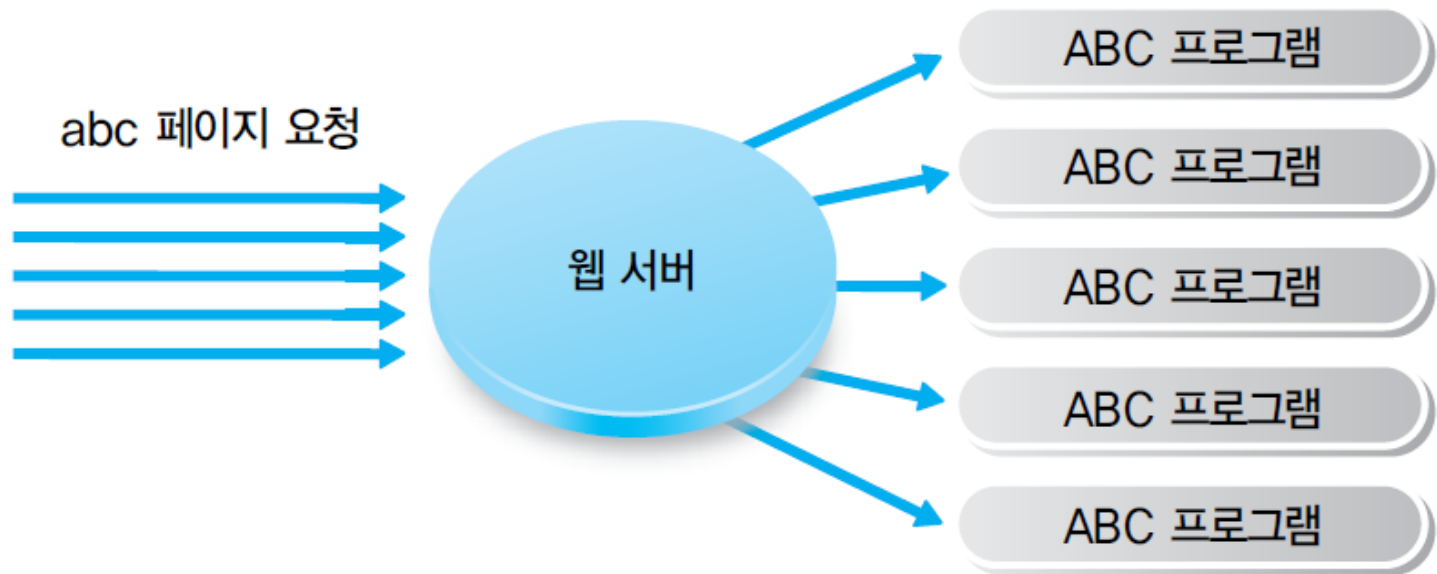
- 웹 브라우저 - 클라이언트이며, 사용자의 작업 창
- 웹 서버 - 웹 브라우저의 요청을 받아들이는 곳
- 웹 애플리케이션 서버(WAS) - 요청된 페이지의 로직 및 데이터베이스와의 연동을 처리하는 부분
- 데이터베이스 - 데이터의 저장소

웹 애플리케이션 처리 방식 및 구현 방식

□ 웹 애플리케이션 처리 방식

▣ CGI(Common Gateway Interface) 방식

- 하나의 요청에 대해 1개의 프로세스가 생성이 되어서 그 요청을 처리한 뒤 종료
- 시스템에 많은 부하를 가져오기 때문에 일부의 UNIX 플랫폼을 제외하고는 CGI 방식을 사용하지 않음



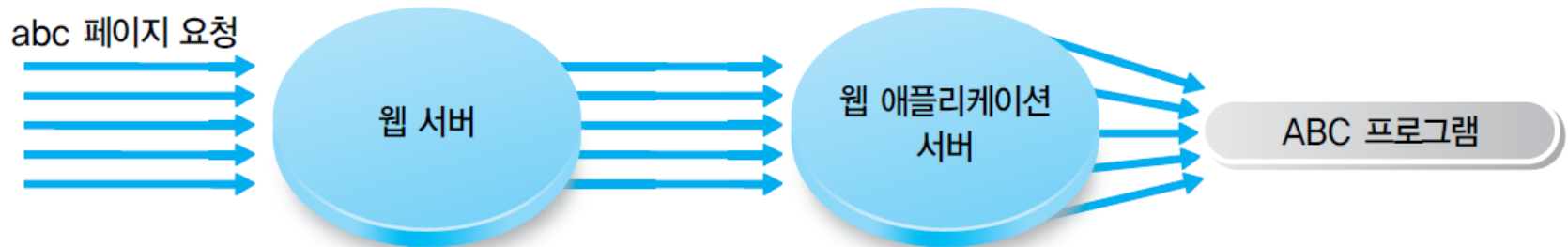
▲ CGI 방식의 구조

웹 애플리케이션 처리 방식 및 구현 방식

□ 웹 애플리케이션 처리 방식

▣ 웹 애플리케이션 서버(Web Application Server) 방식

- 여러 명의 사용자가 동일한 페이지를 요청하여 같은 애플리케이션 프로그램을 처리할 때 1 개의 프로세스만을 할당하고, 사용자의 요청을 스레드(Thread) 방식으로 처리



▲ 웹 애플리케이션 방식의 구조

웹 애플리케이션 처리 방식 및 구현 방식

□ 웹 애플리케이션 구현 방식

▣ 실행 코드 방식

- 미리 컴파일된 실행 프로그램을 사용자가 요청하면, 실행
- CGI 방식에서 사용

▣ 스크립트 코드 방식

- 사용자의 요청이 있을 때, 스크립트 코드를 번역해서 번역된 코드를 실행
- ASP, JSP등의 웹 애플리케이션 서버 방식에서 사용

JSP 및 서블릿의 개요

□ JSP의 개요

▣ 특징

- 객체 지향적
- 플랫폼 독립적
- 네트워크 지향적
- 뛰어난 보안성
- 멀티 스레드 기능
- 친근한 코드

JSP 및 서블릿의 개요

□ JSP의 개요

▣ JSP 페이지의 구조

- 정적페이지 + 동적페이지 = JSP
- 정적페이지 구현 : HTML태그
- 동적페이지 구현 : `<%@ %>`, `<% %>`, `<%= %>` 등의 스크립트 사용

JSP 및 서블릿의 개요

□ JSP 개요

▣ JSP 페이지의 구조

```
<html>
<head>
<meta charset="UTF-8">
<title>JSP 예제</title>
</head>
<body>
<% Date nowDate = new Date();
    SimpleDateFormat dateFormat = new
        SimpleDateFormat("yyyy년MM월dd일");
    String formatDate = dateFormat.format(nowDate);
%>
현재 날짜는 <%= formatDate%> 입니다.</p>
</body>
/html>
```

JSP 및 서블릿의 개요

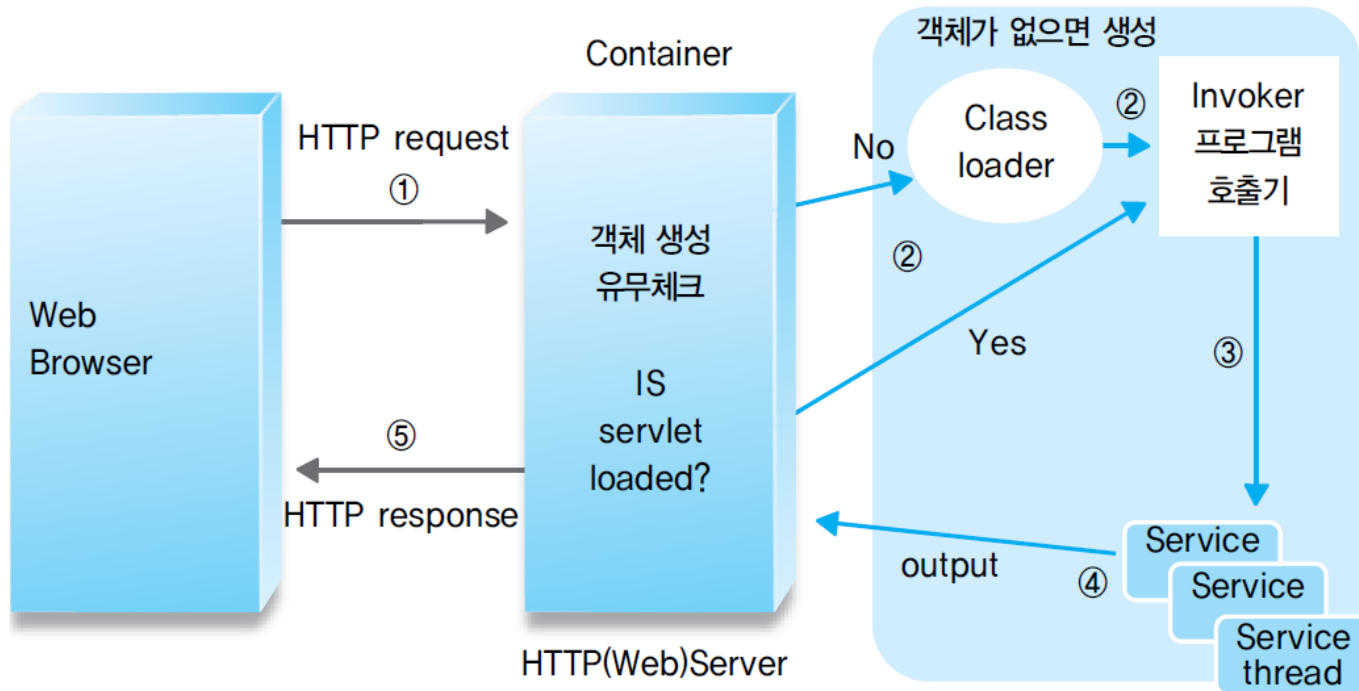
□ 서블릿의 개요

- 멀티 스레딩으로 사용자 요구를 처리하고 가공해서 이에 대한 결과를 내보내는 구조
- JSP 페이지는 화면에 결과를 표시하는 정적인 부분을 담당, 서블릿은 웹 어플리케이션의 흐름을 제어해서 효율적인 웹 사이트를 구성
- 제네릭 서블릿은 `javax.servlet.GenericServlet` 클래스를 상속받음
- HTTP 서블릿은 `javax.servlet.http.HttpServlet` 클래스를 상속받아 생성

JSP 및 서블릿의 개요

□ 서블릿의 개요

▣ HTTP서블릿의 요청과 응답 과정



▲ HTTP 서블릿의 요청과 응답 과정

2. JSP의 개발 환경 설정

- **JDK다운로드 및 설치**
 - ▣ JDK 다운로드하기
 - ▣ JDK 설치하기
 - ▣ 자바 환경변수 설정하기
 - ▣ JDK 설치 확인하기
- **웹 컨테이너 톰캣 다운로드 및 설치**
 - ▣ 톰캣 다운로드하기
 - ▣ 톰캣 설치하기
- **통합환경 이클립스 다운로드 및 설치하기**
 - ▣ 이클립스 다운로드 및 설치하기
 - ▣ 이클립스 실행 및 작업 환경 설정하기

2. JSP의 개발 환경 설정

- **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
 - ▣ JSP 처리과정

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
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_05
PATH	;%JAVA_HOME%\bin

□ 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 컨테이너 톰캣(Tomcat)

- ▣ JSP 웹 애플리케이션을 작성할 수 있는 J2EE 영역을 제공
- ▣ JSP 및 서블릿을 작성하고 실행할 수 있는 웹 컨테이너로 웹 서버와 웹 애플리케이션 서버의 기능도 가짐

□ 톰캣 다운로드

- ▣ <http://tomcat.apache.org>

□ 톰캣 설치

- ▣ 다운로드 받은 톰캣 파일의 압축 해제 및 installer 실행

통합 개발 환경 이클립스 다운로드 및 설

□ 통합 개발 환경 이클립스(eclipse)

- 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

□ 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

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

이클립스에서 웹 애플리케이션 작성

□ 웹 서버 작성

- 설치된 톰캣 서버를 이클립스에서 제어하도록 지정
- 서버 설정
이클립스에서 [File]-[New]-[Other] 메뉴 선택
- [new]창의 [Wizard]에서 [Server]를 선택하고 [next]버튼 선택
- [New Server]창의 [Define a New Server]화면에서 [Select the server type]에서 [Apache]의 [Tomcat v.8.0 Server] 선택 후 [Next] 버튼 클릭
- [Tomcat installation directory]의 [browse]버튼을 클릭하여 톰캣 서버의 위치[C:\Program Files\Apache Software Foundation\Tomcat 8.0]를 지정하고[finish]버튼 클릭

이클립스에서 웹 애플리케이션 작성

□ 동적 웹 프로젝트 작성

- 이클립스에서 애플리케이션은 프로젝트 단위로 프로그래밍
- 웹 어플리케이션은 동적 웹 프로젝트(Dynamic Web Project)로 생성해 실제 환경에 배포
- 동적 웹 프로젝트를 작성해야 웹 페이지를 작성
- [File]-[New]-[Other] 메뉴 선택
[Web]-[Dynamic Web Project] 항목으로 생성
- 생성된 동적 웹 프로젝트는 서버에 추가

이클립스에서 웹 애플리케이션 작성

□ JSP 페이지와 서블릿 작성

▣ JSP 페이지 작성 및 실행

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

▣ 서블릿(Servlet) 작성 및 실행

- 서블릿은 [프로젝트]-[Java Ret] 폴더에 생성
- [Run As]-[Run on Server] 메뉴를 사용해서 JSP페이지를 실행

이클립스에서 웹 애플리케이션 작성

□ 웹 애플리케이션 배포 - WAR 내보내기

- 웹 어플리케이션을 웹상에서 서비스를 하려면, 어플리케이션개발 후 실제로 서비스하는 서버로 배포파일인 WAR파일을 배포
- 프로젝트에서 오른쪽 버튼 [Export]-[WAR file] 메뉴를 선택
- Destination 상자에 War file을 배포한 위치 선택 후 [finish] 클릭
- 배포파일로 만들어진 War file을 [톰캣 홈]-[webapps]폴더에 복사
- 웹 브라우저에 "http://localhost/프로젝트명/"

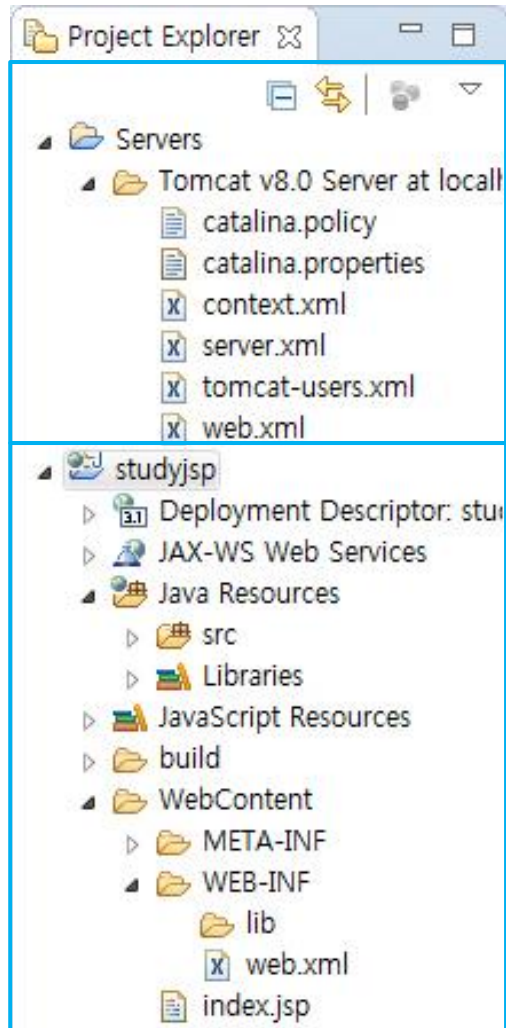
웹 애플리케이션 폴더 구조와 JSP 처리 과정

□ 웹 애플리케이션 폴더 구조

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

웹 애플리케이션 폴더 구조와 JSP 처리 과정

□ 웹 애플리케이션 폴더 구조

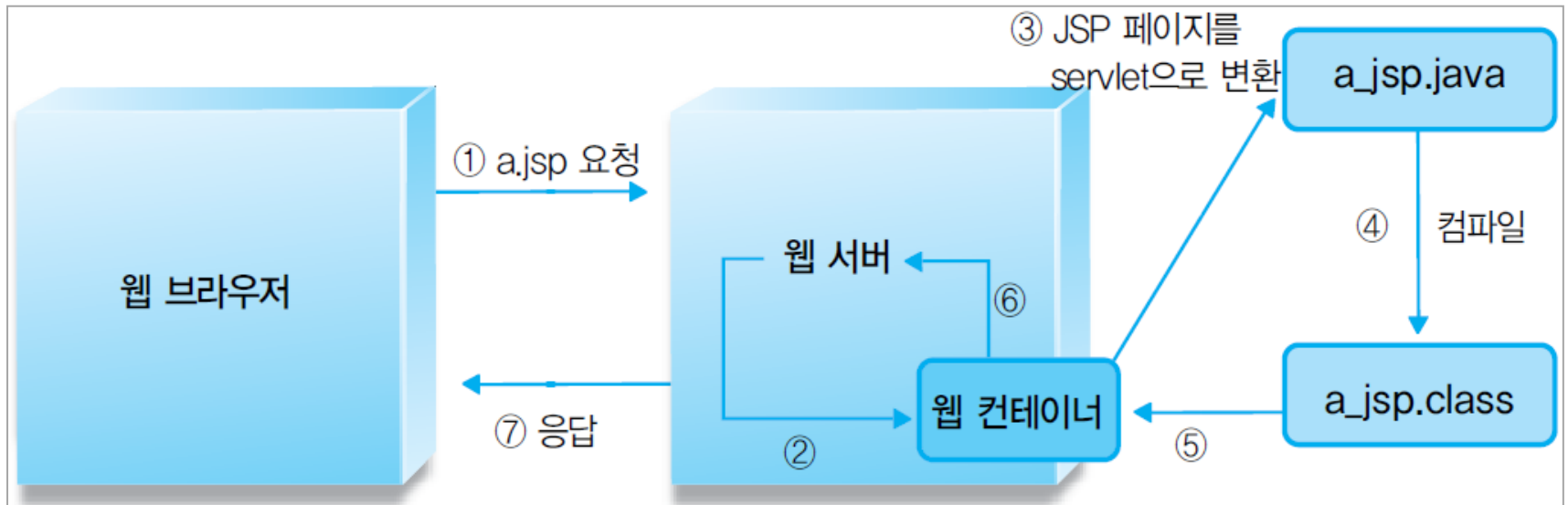


이클립스의 가상 환경
웹 서버의 설정 파일

이클립스의 웹 애플리케이션이
되는 동적 웹 프로젝트

웹 애플리케이션 폴더 구조와 JSP 처리 과정

□ JSP의 처리 과정



▲ 단계별 JSP 페이지의 내부적 처리 과정

웹 애플리케이션 폴더 구조와 JSP 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 것이면 JSP 파일을 서블릿으로 파싱(a.jsp 페이지를 `a_jsp.java`로 변환) -> 이전에 요청되었던 페이지일 경우 ⑥의 단계로 넘어감
- 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 브라우저에 응답