

## [01-31] JSP

### 1. 개요

#### ❖ Servlet과 JSP는 자바를 만든 Sun에서 정한 웹 개발 표준

- 서블릿(Servlet) : 컴파일 방식
- JSP(JavaServerPage) : 스크립트 방식(자동 컴파일)

#### ❖ 서블릿(Servlet) 특징

- 자바 클래스로 작성하며 **HttpServlet** 클래스를 상속해 구현함
- HTTP에 대한 클라이언트의 요청/응답 처리
- 자바 웹 애플리케이션에서 요청을 받아 프로그램의 흐름을 제어하는 용도로 주로 사용
  - MVC 패턴에서 프로그램의 흐름을 제어하는 컨트롤러로 활용됨

#### ❖ JSP

- 자바 기반 서버사이드 스크립트 언어
- 자바의 장점을 그대로 사용
- HTTP에 대한 클라이언트의 요청/응답 처리
- 자바 웹 애플리케이션에서 요청을 처리한 결과 화면을 생성하는 용도로 주로 사용
  - MVC 패턴에서 **View Page**를 작성할 때 활용됨

#### ❖ 웹 컨테이너

- 웹 애플리케이션을 실행할 수 있는 웹 서버를 포함하는 컨테이너
- JSP도 웹 컨테이너에 의해서 **Servlet**으로 변환되어 실행됨
- **Servlet**을 관리하고 실행하기 때문에 서블릿 컨테이너라고 함
- 주요 웹 컨테이너
  - 톰캣(Tomcat) : <http://tomcat.apache.org/>
  - 제티(Jetty) : <http://www.mortbay.org/jetty/>
  - 레신(Resin) : <http://www.caucho.com/>
- 주요 상용 WAS
  - 제우스(JEUS) : <https://www.tmaxsoft.com/kr/solution>
  - 웹로직(WebLogic) : <https://www.oracle.com/kr/java/weblogic/>
  - 웹스피어(WebSphere) : <https://www.ibm.com/kr-ko/products/websphere-application-server>

#### ❖ Java LTS 8, 11, 17

Oracle Java SE Support Roadmap**				
Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Sup
8 (LTS)**	March 2014	March 2022	December 2030*****	Indefinite
9 - 10 (non-LTS)	September 2017 - March 2018	March 2018 - September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	January 2032*****	Indefinite
12 - 16 (non-LTS)	March 2019 - March 2021	September 2019 - September 2021	Not Available	Indefinite
17 (LTS)	September 2021	September 2026****	September 2029****	Indefinite
18 (non-LTS)	March 2022	September 2022	Not Available	Indefinite
19 (non-LTS)	September 2022	March 2023	Not Available	Indefinite
20 (non-LTS)	March 2023	September 2023	Not Available	Indefinite
21 (LTS)	September 2023	September 2028****	September 2031****	Indefinite
22 (non-LTS)***	March 2024	September 2024	Not Available	Indefinite
23 (non-LTS)***	September 2024	March 2025	Not Available	Indefinite
24 (non-LTS)***	March 2025	September 2025	Not Available	Indefinite
25 (LTS)***	September 2025	September 2030	September 2033	Indefinite

<https://www.oracle.com/java/technologies/java-se-support-roadmap.html>

## 2. 자바 웹 개발 환경구축

### 2.1 자바 웹 개발에 필요한 프로그램 다운로드 및 설치

- [JDK 17.xx\(Java SE Development Kit 17\) 이상 다운로드](#)
- [Eclipse IDE for Java EE Developers 다운로드 및 설치](#)
  - 권장 설정



- C: > DeveloperTools 에 Eclipse 압축 풀기 > eclipse (자동생성) -> 디렉토리명 eclipse\_jsp 변경
- eclipse 실행파일 > 우클릭, 바탕화면 바로가기 > 아이콘명을 eclipse\_jsp 변경
- 실행 > D: > JSPStudy 생성 > D:\JSPStudy 로 설정, 체크박스 체크 > Launch
- [서블릿 컨테이너 Tomcat 9.0.xx 다운로드](#) > Binary Distributions > [64-bit Windows zip \(pgp, sha512\)](#)
  - apache 관련 프로젝트 모음 : <https://projects.apache.org/projects.html>

### 2.2 이클립스 환경 설정 및 톰캣 서버 등록하기

1) 이클립스에서 프로젝트를 만들기 전에 이클립스의 기본 설정과 톰캣 서버를 등록해 보자.

Window > Preferences > General > Appearance > Colors and Fonts > Basic > Text Font

Window > Preferences > General > WebBrowser > Use external web browser > Chrome

Window > Preferences > General > Workspace > Text file encoding : Other UTF-8

최근의 이클립스 배포판은 OpenJDK가 포함되어 배포되기 때문에 별도로 JDK 를 설치하지 않아도 바로 자바 개발을 진행할 수 있다. 참고로 “Eclipse IDE for Enterprise Java and Web Developers 2023-12 (4.30.0)” 버전에 포함된 JDK는 OpenJDK-17.0.x 버전이다.

Window > Preferences > General > Java > installed JREs 에서 JDK 17 설치 확인 가능

2) 앞에서 설치한 톰캣을 이클립스와 연동해 사용하기 위해서 서버 실행환경으로 등록해 보자.

Window > Preferences > General > Server > Runtime Environments > Add > Apache > Apache Tomcat v9.0 선택 > Browse... > 톰캣 설치 폴더(C:\DeveloperTools\apache-tomcat-9.0.85) > Finish

Window > Preferences > General > Web > CSS Files > ISO 10646/Unicode(UTF-8)

Window > Preferences > General > Web > HTML Files > ISO 10646/Unicode(UTF-8)

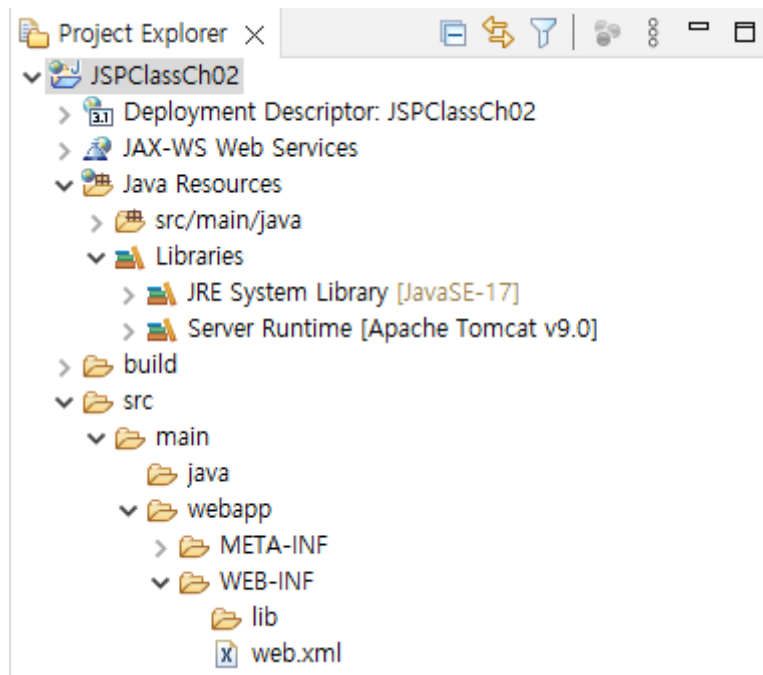
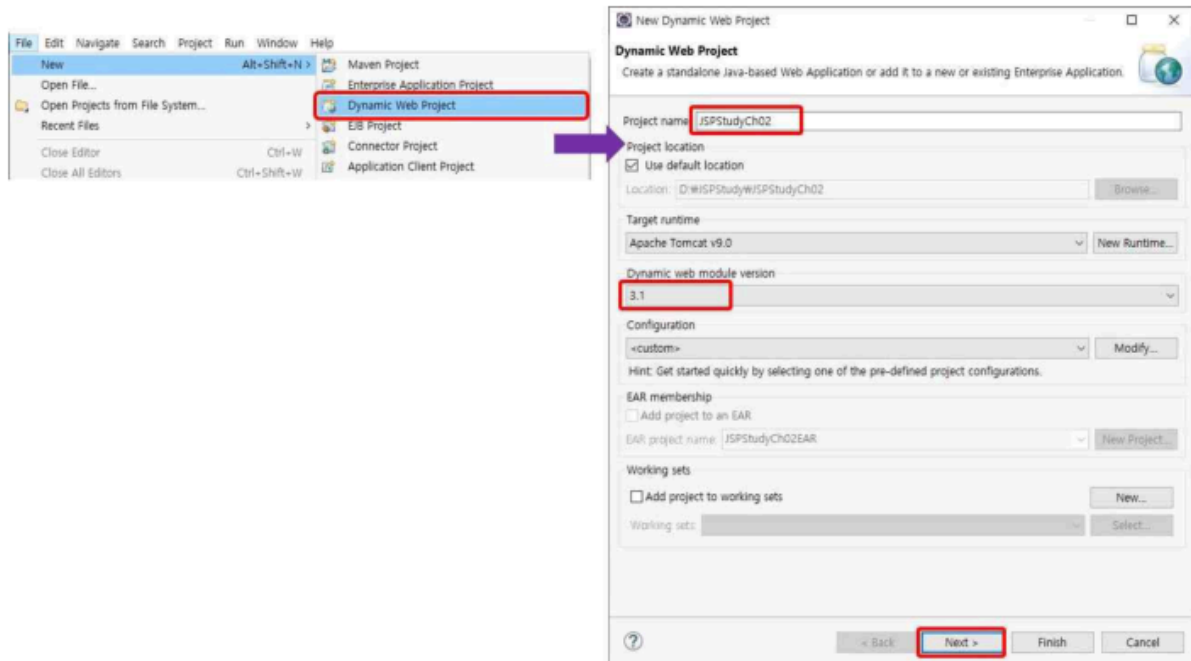
Window > Preferences > General > Web > JSP Files > ISO 10646/Unicode(UTF-8)

### 2.3 오라클 포트 변경

만약 컴퓨터에 OracleXE가 설치되어 실행 중 이라면 오라클이 8080포트를 기본 httpport로 사용하기 때문에 변경이 필요

## 2.4 프로젝트 만들기

File > New > Dynamic Web Project > Project name(JSPClassCh02), Dynamic web module version(3.1) > next > next > web.xml은 배포파일을 의미, 체크박스 체크 > Finish



- Java Resource 에서 JRE 와 Tomcat 이 확인됨
- WEB-INF 는 외부에서 확인불가, 여기에 파일 넣으면 안됨
- META-INF 에도 파일 넣으면 안됨
- JSP 파일은 webapp 에 파일 넣어야 Tomcat과 연동되어 실행됨

## 2.5 이클립스에서 jsp 페이지 만들기

1) 새 파일 생성 : webapp > new or webapp > new > other > web > jsp

2) JSP 기초 문법

- <% 로 시작해서 %> 로 끝남
- @는 지시자를 의미
- scriptlet - java 코드를 작성하는 곳

3) 전체 소스코드

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

<%
//scriptlet - 자바코드 작성하는 곳
Calendar today = Calendar.getInstance();
    System.out.println(today.get(Calendar.YEAR));
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%= today.get(Calendar.YEAR) %> 년
<%= today.get(Calendar.MONTH)+1 %> 월
<%= today.get(Calendar.DAY_OF_MONTH) %> 일입니다.
</body>
</html>
```

<%@ page import="java.util.\*" %> : import 가 없으면 Java 코드가 에러 발생

4) JSP 실행 : ctrl + F11 (실행) > Tomcat 9.0 선택 > Finish > 새 창에 결과 출력

## 2.6 이클립스에서 서블릿 만들기

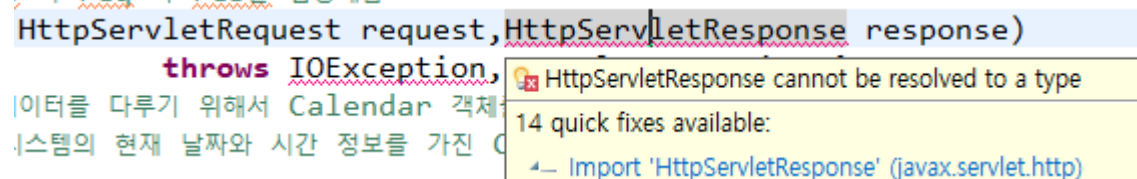
1) Source Code Distributions : 설치해야 Tomcat 함수의 설명을 확인할 수 있다.

- <https://tomcat.apache.org/download-90.cgi> > Source Code Distributions > zip 다운 > attached file > outer link > 다운받은 압축파일 지정

2) 이클립스에서 서블릿 파일 생성 : Java 생성과 동일하므로 해당 루트로 파일을 생성한다. 또한 패키지는 관례적으로 전부 소문자, 역순으로 작성한다.

- Java Resource > src/main/java > new > package > com.jspstudy.ch02.servlet
- 생성된 파일 우클릭 > new > class > name : TodayServlet > Finish

3) 첫 구문 작성시 에러 문구가 발생하는데 하나씩 클릭하면 자동 import 가 생성된다.



4) 전체 코드

```
package com.jspstudy.ch02.servlet;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Calendar;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
//서블릿은 HttpServlet 을 상속받아야 함
@WebServlet("/today")
public class TodayServlet extends HttpServlet {
    @Override
    public void doGet(
//tomcat 이 req 와 res를 담당해줌
        HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        Calendar today = Calendar.getInstance();
        /* 웹 브라우저에 응답할 문서의 형식과 문자 셋 지정
        * 문서의 형식과 문자 셋을 잘못지정하면 한글과 같은 문자는 깨져서 표시된다.
        * 응답 객체의 문서 형식과 문자 셋은 응답 객체의 스트림에 접근하기 전에 설정해야
        * 제대로 동작한다. 만약 응답 객체의 스트림에 접근한 후에 응답 객체의 문서 형식과
        * 문자 셋을 지정하면 영문, 숫자 등은 제대로 표시되지만 한글은 깨져서 표시된다.
        */
        response.setContentType("text/html; charset=utf-8");
        // 웹 브라우저로 응답 결과를 보내기 위해 응답 객체를 통해 스트림객체를 구한다.
        PrintWriter out = response.getWriter();
        /* 다음과 같이 응답 객체에 연결된 스트림의 출력 메소드를 이용해 HTML 문서 형식에
        * 맞게 내용을 작성하면 이 데이터가 브라우저로 전달되어 브라우저 화면에 표시된다.
        * 브라우저는 텍스트와 html 문법만 알기 때문에 직접 작성해줘야 한다.
        */
    }
}
```

```
out.println("<html>");
out.println(" <head><title>오늘의 날짜</title></head>");
out.println(" <body>");
out.print("오늘은 " + today.get(Calendar.YEAR) + "년 ");
out.print((today.get(Calendar.MONTH) + 1) + "월 ");
out.print(today.get(Calendar.DAY_OF_MONTH) + "일 입니다.");
out.println(" </body>");
out.println("</html>");
// 작업이 끝나면 스트림을 닫는다.
out.close();
}}
```

#### 4-2) 코드 부가 설명

jsp 는 자동 맵핑되지만, servlet 은 servlet맵핑에 의해 호출된다.  
@WebServlet("/today") 의 컨텍스트 루트(/)는 JSPClassCh02 이다.