



# JSP 기본 구조 및 문법

# 실습환경구성

## ❖ JDK설치

- [www.oracle.com](http://www.oracle.com) 사이트 접속하여 Java SE JDK를 다운로드 받아 설치

## ❖ Tomcat 설치

- [www.apache.org](http://www.apache.org) 사이트 접속하여 tomcat 8.0버전 다운로드
- 32bit/64bit installer

## ❖ Eclipse Web 개발환경 설정

- [www.eclipse.org](http://www.eclipse.org) 사이트 접속하여 java EE Developers 다운로드
- 적당한 곳에 압축해제 및 Workspace 연동

## ❖ JSP 코드 작성 및 테스트(EX01.jsp)

- 실행 확인 후 Servlet 코드 확인
- 생성된 Servlet 코드 경로

Workspace ▶ .metadata ▶ .plugins ▶ org.eclipse.wst.server.core ▶ tmp0 ▶ work ▶ Catalina ▶ localhost ▶

- 위 경로 안에 자신이 만든 프로젝트 이름 폴더에 들어가면  
jsp파일과 같은 이름의 java파일이 있다.
- 해당 jsp페이지를 tomcat이 Servlet으로 변환한 java코드이다.
- 작성된 Servlet코드를 문서편집기로 열어 내용 확인

# JSP[Java Server Page]

## ❖ JSP

- JSP는 Servlet을 사용하여 웹 페이지를 개발하기 위한 기술
- HTML코드에 JAVA코드를 삽입하는 방식
- 웹 어플리케이션이 JSP페이지의 내용을 Servlet으로 변환 후 실행

## ❖ 특징

- Beans Java Component 사용가능
- 최초 컴파일 후 메모리에서 처리
- 자바의 모든 기능 사용 가능(무한한 확장성)
- 사용자 태그 및 JSTL사용 등 다양한 태그 사용가능
- 여러 실행환경에서 동작가능(호환성)

# JSP 기본 문법(주석)

## ❖ 주석

- <!-- HTML주석 -->
- <%-- JSP주석 --%>
- // JAVA 한 줄 주석
- /\* 여러 줄 주석 \*/

```
<!-- JSP에서 사용하는 HTML주석 -->  
<!-- HTML주석 : <%=comment%> --><br></br>  
<%-- JSP주석 : <%=comment%> --%><br></br>
```

```
<%  
//자바주석  
/*  
범위 주석  
*/  
%>
```

# JSP 기본 문법(EX02.jsp)

## ❖ 선언문(declaration)

- JSP 페이지에서 멤버 변수, 멤버 메서드를 선언할 때 사용
- `jspInit()`, `jspDestroy()` 등을 재정의(overriding) 할 때 사용
- **<%! 선언문; %>**

## ❖ 스크립틀릿(scriptlet)

- JSP 페이지에서 작성된 자바코드를 지원
- `_jspService()`에 원하는 자바코드를 JSP페이지에 정의하도록 지원
- 스크립틀릿에 선언된 변수는 지역변수로 설정됨
- 메서드 호출가능(선언불가)
- **<% 자바코드 %>**

## ❖ 표현식(expression)

- JSP 페이지에서 직접 클라이언트로 출력될 내용을 표시하는 부분
- `_jspService()`에 삽입된다. 표현식에서 `;`(세미 콜론) 사용 안 함
- **<%= 표현식 %>**

# JSP Life Cycle(EX03.jsp)

## 1. 클라이언트 요청

- 클라이언트가 웹 브라우저에서 웹 서버로 jsp페이지 요청

## 2. 요청 분석 및 페이지 소스 컴파일 후 초기화(jspInit())

- 클라이언트가 요청한 페이지에 대한 서블릿 확인(있으면 \_jspService()호출)
- 없으면 컴파일하여 서블릿 생성 후 \_jspService()호출하여 응답처리

## 3. 요청 처리(\_jspService())

- 클라이언트의 요청을 처리해주는 메소드

## 4. 객체 해제(jspDestroy())

- 서버의 페이지 내용이 변경된 후 다시 요청 시 jspDestroy()호출하여 현재 객체를 정리(해제) 후 jspInit()다시 호출하여 \_jspService()를 통해 요청 처리

# Directive[지시어]

❖ JSP 페이지를 처리하는 방법에 대한 정보를 알려주는 역할

❖ page

- 페이지 관련 환경
- 주로 문자 인코딩, 응답 페이지의 콘텐츠 타입 등을 정의

```
<%@ page 속성1="속성값1" 속성2="속성값2"... %>
```

❖ language

- JSP 페이지에서 사용되는 스크립트 언어 지정(java)
- <%@ page language="java"%>

❖ import

- import 할 패키지 명시
- <%@ page import="java.util.ArrayList"%>

# Quiz

➤ 다음 code를 import 디렉티브를 사용하여 수정하기(quiz.jsp)

```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
7 <title>Calendar 클래스 사용</title>
8 </head>
9 <body>
10 <%
11     java.util.Calendar cal = java.util.Calendar.getInstance();
12 %>
13 오늘은
14     <%= cal.get(java.util.Calendar.YEAR) %> 년
15     <%= cal.get(java.util.Calendar.MONTH) + 1 %> 월
16     <%= cal.get(java.util.Calendar.DATE) %> 일
17 입니다.
18 </body>
19 </html>
```



# Directive

## ❖ info

- JSP 페이지의 정보를 관리
- `getServletInfo()` 를 사용하여 확인(EX05.jsp)

## ❖ errorPage

- JSP 페이지에서 에러가 발생할 경우 이동할 에러페이지 경로 지정(EX06.jsp)
- EX06.jsp를 실행하면 에러코드 500에 대한 기본 에러페이지가 보여진다.
- 다음 경로에 **error.jsp** 파일을 작성하여 에러코드 500의 에러 페이지를 변경
- 테스트 경로 : `/basic/error/error.jsp`

## ❖ isErrorPage

- 이 페이지가 에러 페이지인지를 표시(**error.jsp**)
- IE는 범용 에러에 대해 Tomcat에서 처리(간단히 Chrome으로 테스트 가능)
- `web.xml` 파일에 에러에 대한 페이지의 우선순위를 지정해야 됨.

# Directive

## ❖ contentType

- JSP 페이지의 MIME, 문자 인코딩 타입을 지정(EX07.jsp)

## ❖ pageEncoding

- 웹 컨테이너가 JSP페이지를 분석 할 때 어떤 인코딩으로 작성되었는지 검사하여 JSP페이지의 문자를 읽는다.
- contentType의 charset과 pageEncoding의 charset이 서로 다를 수 있다.

## ❖ include

- 현재 JSP페이지에 사용할 파일을 포함
- text, html, jsp, url 등을 지정할 수 있다. (url을 지정할 경우 상대경로 사용)  
(EX08\_top.jsp, EX08\_bottom.jsp, EX08\_include.jsp)

# 에러처리

## ❖ 자주 보이는 에러코드

- 404(페이지를 찾을 수 없음)
- 500(오타로 인한 처리 에러)

## ❖ web.xml에 코드를 지정하여 에러상황에 대한 페이지를 지정

- web.xml 파일을 수정(아래 그림 참조)
- /basic/error/EX09\_error404.jsp
- /basic/error/EX09\_error500.jsp

```
11 </welcome-file-list>
12 <error-page>
13     <error-code>404</error-code>
14     <location>/sample/error/EX09_error404.jsp</location>
15 </error-page>
16 <error-page>
17     <error-code>500</error-code>
18     <location>/sample/error/EX09_error500.jsp</location>
19 </error-page>
```

- 작성 후 실행(EX10.jsp)
  - EX10.jsp 실행 (500에러)
  - EX10.jsp 의 에러페이지 디렉티브 주석 해제 후 실행 (404에러)