



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음

Chapter 05. JSP 기본 문법

목차

1. 주석
2. 지시어
3. 액션
4. 선언과 표현식
5. 스크립트릿

02. 지시어

■ 지시어(Directives)란?

- 지시어(Directives)는 JSP 파일의 속성을 기술하는 JSP 문법
- JSP 컨테이너에게 해당 페이지를 어떻게 처리해야 하는지 전달하기 위한 내용을 담고 있다.
- 지시어는 크게 page, include, taglib으로 나눌 수 있으며, 각각에서 다루는 속성이 다르다.

1. page 지시어

- page 지시어는 현재 JSP 페이지를 컨테이너에서 처리하는 데 필요한 각종 속성을 기술하는 부분
- 보통 JSP 페이지 맨 앞에 위치함.

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

- 여러 줄에 나누어 작성할 수도 있음.

```
<%@ page contentType="text/html;charset=UTF-8"
import="javax.sql.*, java.util.*" errorPage="error.jsp"%>
<%@ page import="java.util.*" %>
```

02. 지시어

- 이클립스 개발 도구를 이용해 jsp 파일을 생성하는 경우 기본적인 page 지시어는 자동 생성됨

```
scriptet1.jsp scriptet2.jsp scriptet3.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
```

- 필요에 따라 속성을 수성/추가해 사용

[표 5-1] page 지시어 속성

속성	설명	기본 설정 값
language	스크립트 언어의 유형을 정한다.	java
import	JSP 내에서 사용할 외부 자바 패키지나 클래스의 불러오기(import)를 정한다.	-
session	세션의 사용 유무를 정한다.	true
buffer	버퍼의 크기를 정한다.	8KB
autoFlush	버퍼의 내용을 자동으로 비운다.	true
isThreadSafe	단일 스레드 모델을 사용함으로써 동시성 제어 여부를 정한다.	true
info	JSP 페이지에 대한 설명이다.	-
errorPage	현재 페이지에서 오류가 발생할 경우 호출될 페이지를 지정한다.	-
isErrorPage	오류만을 처리하는 페이지로 지정한다.	false
contentType	MIME 형식 지정 및 캐릭터셋을 설정한다.	text/html; charset=ISO-8859-1
pageEncoding	contentType과 동일한 기능을 한다.	ISO-8859-1
extends	현재 JSP 페이지를 특정 클래스를 상속한 클래스로 작성한다.	[표준스펙] javax.servlet.jsp.HttpJspPage [톰캣구현] org.apache.jasper.runtime.HttpJspBase

02. 지시어

■ language

- JSP 스크립트릿 부분에서 사용할 언어를 지정하는 속성이다. 기본 값은 java로 생략할 수 있다.

■ import

- import 는 JSP 스크립트 부분에서 자바 클래스를 사용하는 경우 해당 클래스의 패키지에 대한 import 설정으로 기본적으로 자바에서와 동일하다.
- 다만 패키지 import 구분을"," 을 이용하거나 라인 단위로 작성해야 한다.
- 다음은 page 지시어에서 import 속성의 다양한 사용 예이다.

```
<%@ page import="java.sql.*, java.util.*" %>
```

```
<%@ page import="java.sql.*" %>
```

```
<%@ page import="java.util.*" %>
```

```
<%@ page  
    import="java.sql.*,  
    java.util.*"  
%>
```

02. 지시어

■ session

- 세션은 웹 브라우저와 웹 서버가 지속적인 클라이언트 인식을 위해 필요한 정보를 임시로 저장해두는 방법
- 주로 웹 사이트에 로그인하거나 쇼핑몰에서 장바구니 등을 구현할 때 사용된다.
- 기본 값이 true(세션을 사용한다)이므로, 일부러 사용을 제한할 목적이 아니라면 별도로 설정하지 않아도 됨.
- 세션과 관련한 자세한 내용은 6장에서 다룸.

```
<%@ page session="true" %>
```

■ buffer

- JSP 페이지 데이터를 출력하기 위한 JspWriter 즉 out 내장객체의 버퍼 크기를 지정.
- 기본값은 8KB 이고 JSP 페이지에 동적으로 많은 내용이 포함될 경우 버퍼 크기 조정이 필요할 수도 있으나 일반적으로는 변경하지 않아도 됨.

02. 지시어

■ autoFlush

- autoFlush는 버퍼를 자동으로 비울 것인지를 지정하는 속성으로, 기본 값은 true다.
- 버퍼 속성에 지정되어 있는 크기만큼 버퍼를 유지하고 있다가 버퍼가 다 차면 자동으로 전송한다.

```
<%@ page autoFlush="true" %>
```

■ isThreadSafe

- 기본적으로 서블릿은 스레드로 동작하기 때문에 스레드로 인한 동기화 문제를 해결하기 위한 옵션임.
- 기본값은 true로, 일반적으로 false로 설정하는 경우는 거의 없다.

```
<%@ page isThreadSafe="true" %>
```

■ info

- 해당 JSP에 대한 간단한 설명으로 저작권이나 작성일 등 간단한 정보 기술에 사용.

```
<%@ page info="JSP Example" %>
```

02. 지시어

■ **errorPage, isErrorPage**

- 두 속성은 jsp파일의 오류 처리를 위한 것으로, errorPage는 현재 페이지에 오류 발생시 호출할 페이지를 지정하는 속성이고 isErrorPage는 오류 처리를 위한 전용 페이지임을 알리는 속성이다.
- errorPage지정을 통해 보다 효과적으로 페이지 오류를 관리할 수 있다.
 - errorPage : 일반적인 JSP 파일에 사용

```
<%@ page errorPage="오류_처리_파일.jsp" %>
```

- isErrorPage : 오류 처리 파일에만 사용

```
<%@ page isErrorPage="true" %>
```


오류페이지 예시

HTTP Status 500 - Something went horribly wrong.

type Exception report

message Something went horribly wrong.

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
java.lang.RuntimeException: Something went horribly wrong.  
    TestServlet.doGet(TestServlet.java:29)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:621)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)  
    org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

note The full stack trace of the root cause is available in the Apache Tomcat/7.0.50 logs.

Apache Tomcat/7.0.50



식품의약품안전처
MINISTRY OF FOOD AND DRUG SAFETY

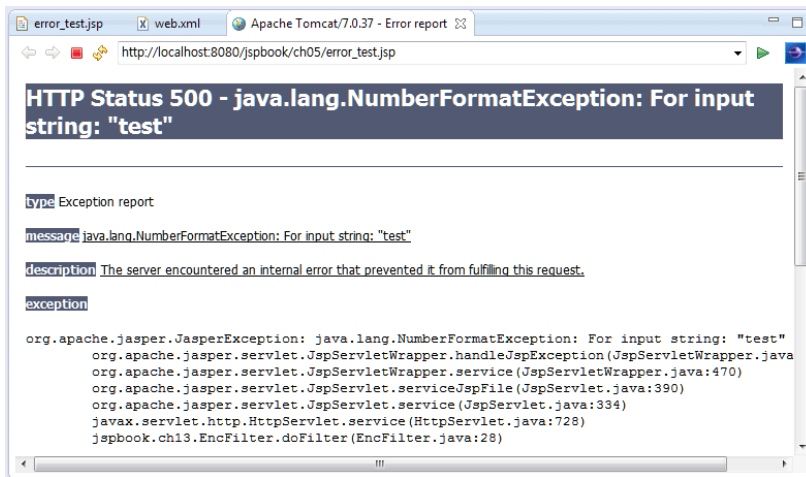
“ 죄송합니다.
선택한 페이지를 찾을 수 없습니다. ”

검색중인 페이지가 제거되었거나 이름이 변경되어
일시적으로 사용할 수 없습니다.
홈페이지를 열고, 원하는 정보가 있는 링크를 다시 검색해 주세요.

메인페이지 바로가기

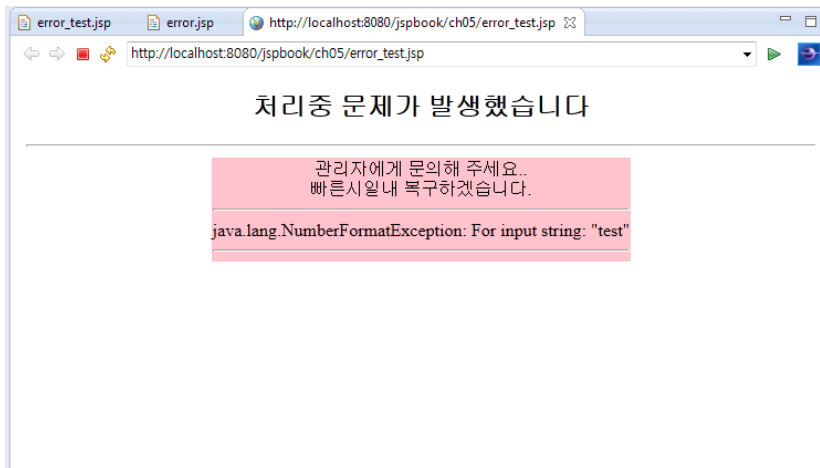
02. 지시어

- [실습] 런타임 오류를 발생시키는 파일(error_test.jsp) - 교재 p.168 참고



[그림 5-9] 일반적인 오류 화면

- [실습] 오류 처리 전용 파일(error.jsp) - 교재 p.169 ~ 170 참고



[그림 5-10] 사용자 정의 오류 페이지 사용

[예제 5-2] error_test.jsp

error_test.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <TITLE>Error Test</TITLE></HEAD>
7 <BODY><H2>Error 테스트 페이지</H2>
8 <%
9 String str = "test";
10 int i = Integer.parseInt(str);
11 %>
12 </BODY>
13 </HTML>
```

[예제 5-3] error.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isErrorPage="true"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html
4 <HTML>
5 <head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 </head>
7 <body>
8 <center>
9   <H2>처리중 문제가 발생했습니다</H2>
10  <HR>
11  <table>
12  <tr bgcolor="pink"><td>
13    관리자에게 문의해 주세요..  
<BR>
14    빠른시일내 복구하겠습니다.
15  <HR>
16  <%= exception%>
17  <HR>
18  </td>
19  </tr>
20  </table>
21  </center>
22  </body>
23  </HTML>
```

[예제 5-2] error_test.jsp (revisited)

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" errorPage = "error.jsp" %>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/1
4 <html>
5 <head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <TITLE>Error Test</TITLE></HEAD>
7 <BODY><H2>Error 테스트 페이지</H2>
8 <%
9   String str = "test";
10  int i = Integer.parseInt(str);
11  %>
12 </BODY>
13 </HTML>
```

02. 지시어

■ contentType

- `<%@ page contentType="text/html; charset=UTF-8" %>`

■ pageEncoding

- pageEncoding은 컨테이너에서 처리할 JSP 파일의 인코딩을 설정.

```
<%@ page pageEncoding="UTF-8" %>
```

- JSP 2.0 스펙에 추가된 속성으로, 이전 버전을 지원하는 컨테이너의 경우에는 사용할 수 없다.

02. 지시어

■ page 지시어와 JSP의 한글 처리 (교재 p. 163)

- page 지시어에서 중요한 부분 중 하나는 한글 처리 부분임.
- JSP 에서는 다음과 같이 3단계로 캐릭터셋을 설정함.

[표 5-2] JSP 에서의 캐릭터셋 관련 설정 항목

우선순위	설정 방법	비고
1	server.xml 파일에서 정의한다.	Connector 설정에 URI 요청에 대한 캐릭터셋을 지정할 수 있다. 작성한 소스를 다른 서버에서 실행할 경우, 해당 서버 설정에 따라 한글이 깨질 수도 있다.
2	각 JSP 파일, page 지시어의 pageEncoding 속성에서 정의한다.	페이지 단위의 설정이므로 애플리케이션 설정과 무관하게 한글을 처리할 수 있다. 단, 서버가 JSP 2.0 스펙을 지원하지 않는 예전 버전이라면 pageEncoding 속성을 인식하지 못해 한글이 깨질 수 있다.
3	각 JSP 파일, page 지시어의 contentType 속성에서 정의한다.	클라이언트에 대한 응답 결과를 지정하는 부분으로 브라우저에게 전달할 캐릭터셋을 지정할 수 있다.

- 위 설정에서 캐릭터셋 설정을 찾지 못할 경우 ISO8859-1을 적용한다.
- 페이지 지시어에 다음과 같이 한글 속성 설정
 - pageEncoding="UTF-8", contentType="text/html; charset=UTF-8"

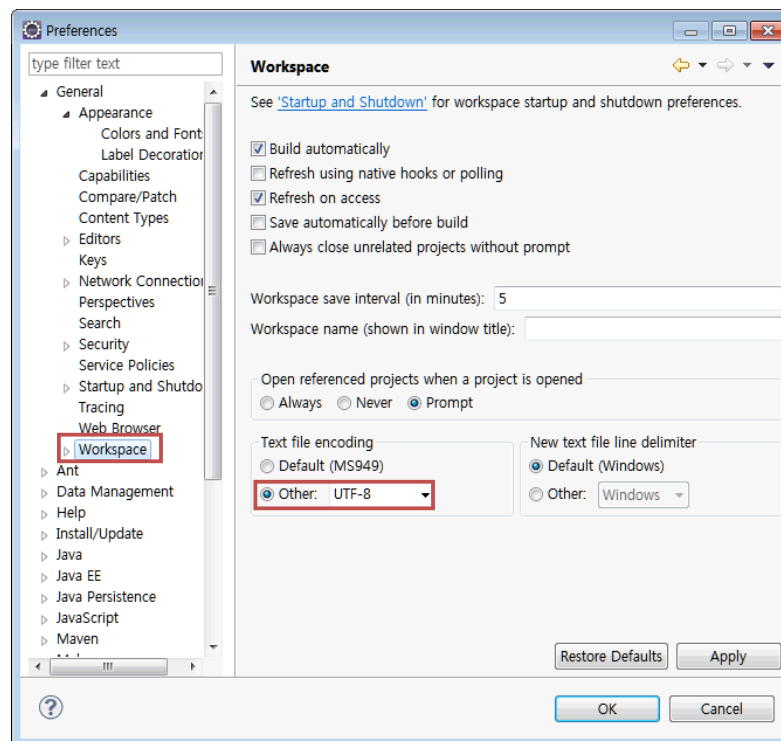
03. 이클립스 기본 환경 설정 (p. 69)

■ 텍스트 인코딩 설정하기

- 프로그램 개발 시 한글 사용은 여러 문제를 발생시키는 원인이 됨.
- 다국어를 지원하는 프로그램 개발을 위해서는 유니코드 기반인 UTF-8 의 사용이 권장됨
 - 이전에는 EUC-KR이나 ISO-8859-1
- 안드로이드 등 모바일 개발에도 유니코드가 기본임

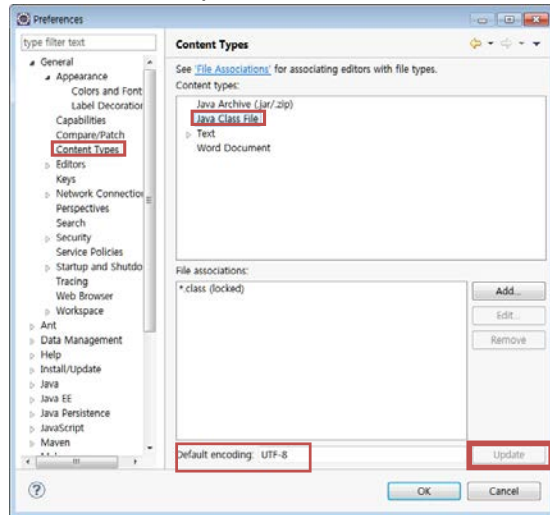
❶ [General] → [Workspace] → Text file encoding 항목을 Other 로 변경한 뒤 UTF-8로 설정

[그림 2-23] Workspace 인코딩 설정



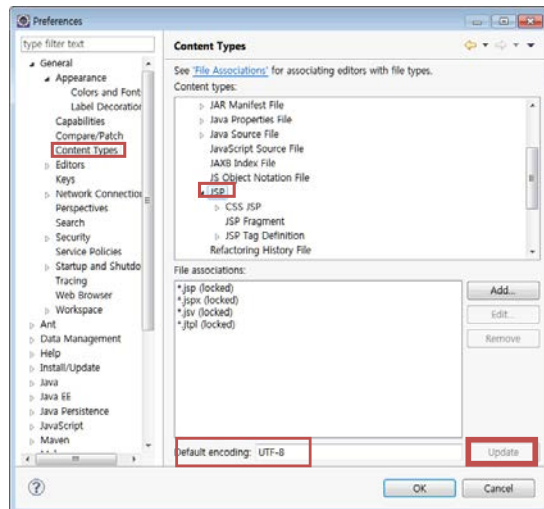
03. 이클립스 기본 환경 설정 (cont'd)

- ② 자바 클래스 인코딩 설정 : [General] → [Content Types] → Java Class File에 대한 Default encoding 값을 UTF-8 로 입력 후 <Update> 버튼



[그림 2-24] Java 인코딩 설정

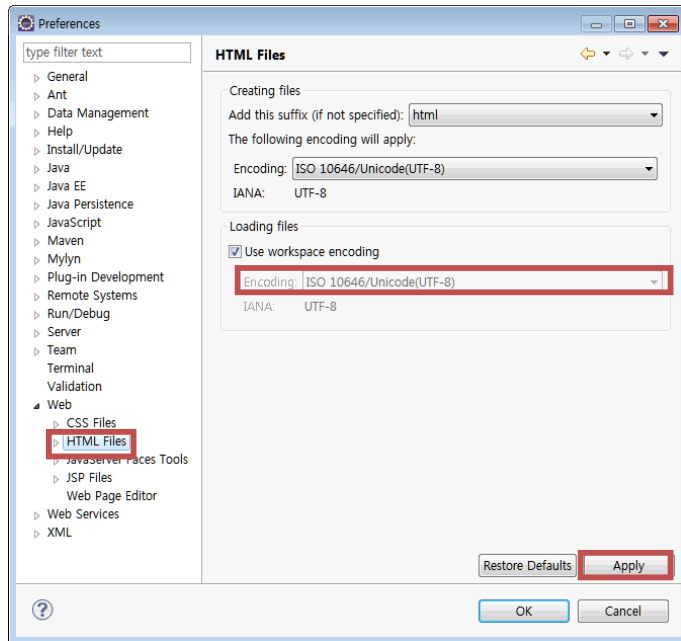
- ③ JSP 파일 인코딩 설정 : [General]→[Content Types] → [Text] → JSP 항목을 UTF-8 로 변경 후 <Update> 버튼



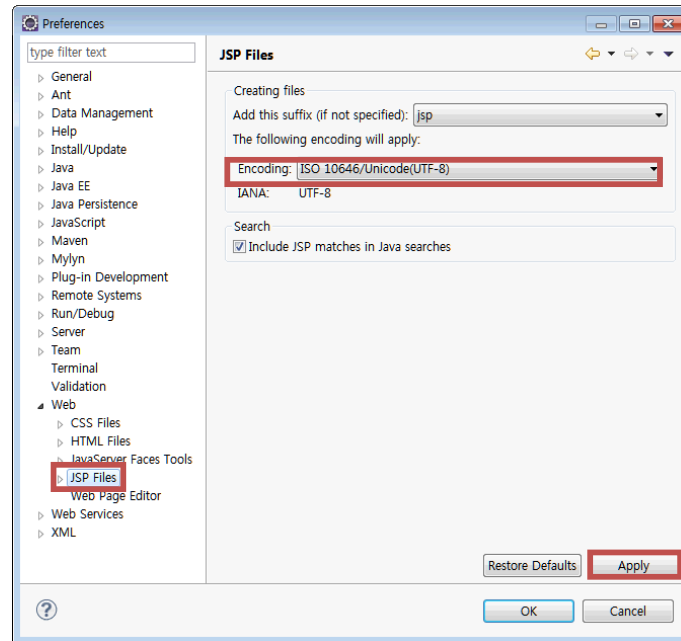
[그림 2-25] JSP 인코딩 설정

03. 이클립스 기본 환경 설정 (cont'd)

- ④ HTML / JSP 코드 인코딩 설정 : [Web] → [HTML Files] 항목을 ISO 10646/Unicode(UTF-8)로 변경
- ⑤ HTML / JSP 코드 인코딩 설정 : [Web] → [JSP Files] 항목을 ISO 10646/Unicode(UTF-8)로 변경



[그림 2-26] HTML Files 인코딩 설정



[그림 2-27] JSP Files 인코딩 설정

02. 지시어

■ contentType

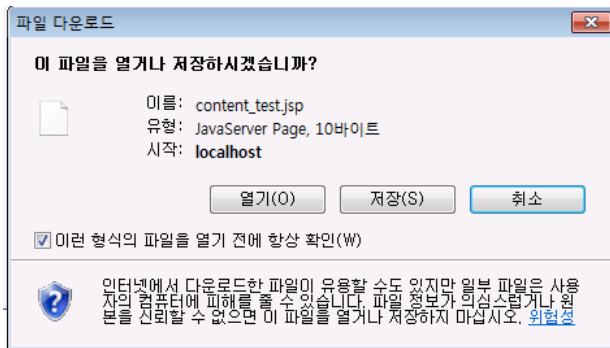
- 현재 JSP 페이지를 클라이언트에서 처리하기 위한 콘텐츠 유형을 지정하는 부분.
- 윈도우에서 파일 확장자(.doc, .hwp 등)에 따라 연결 프로그램이 동작하는 것과 마찬가지로 웹 브라우저에서도 contentType에 따라 전달되는 내용을 어떻게 처리할지 결정할 수 있다.

```
<%@ page contentType="text/html" %>
```

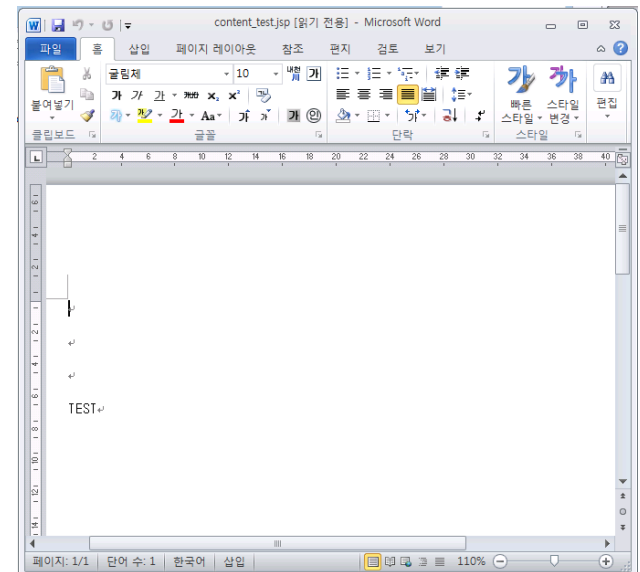
- text/html 이 아니라 application/msword 로 지정할 경우 브라우저는 서버가 전달하는 콘텐츠를 MS-Word 문서로 인식해 처리할 것은 사용자에게 요청함.

```
<%@ page contentType="image/x-rgb" %>
```

```
<%@ page contentType="application/msword" %>  
TEST
```



[그림 5-11] 파일 저장을 묻는 화면

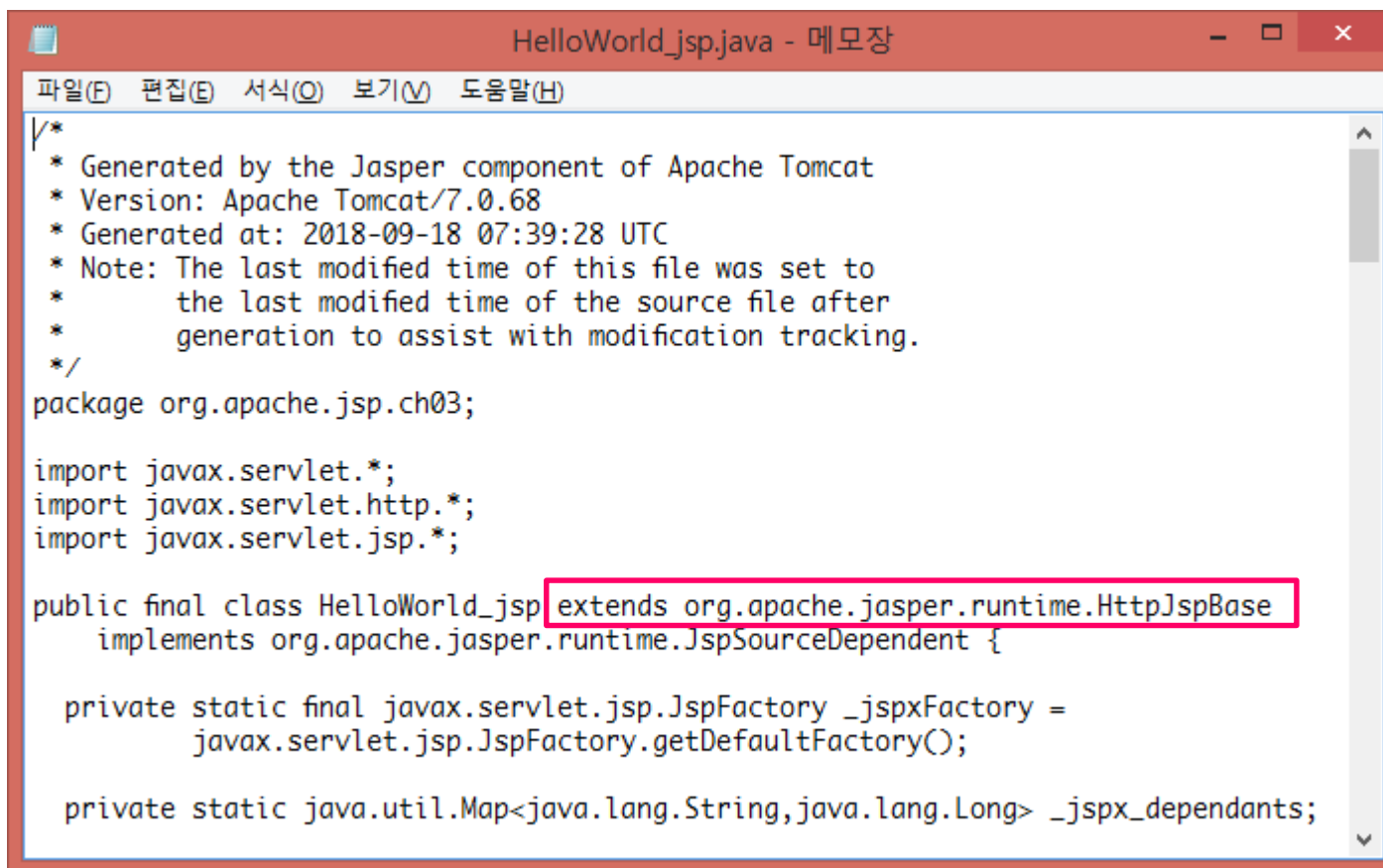


[그림 5-12] MS 워드 실행 화면

02. 지시어

■ extends

- JSP가 서블릿으로 변환될 때 상속받을 슈퍼클래스를 지정할 수 있는 부분이다.
- 일반적으로는 사용할 일이 없으며 톰캣의 경우 org.apache.jasper.runtime.HttpJspBase 클래스를 상속받게 된다.



```

HelloWorld_jsp.java - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
/*
 * Generated by the Jasper component of Apache Tomcat
 * Version: Apache Tomcat/7.0.68
 * Generated at: 2018-09-18 07:39:28 UTC
 * Note: The last modified time of this file was set to
 *       the last modified time of the source file after
 *       generation to assist with modification tracking.
 */
package org.apache.jsp.ch03;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class HelloWorld_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    private static final javax.servlet.jsp.JspFactory _jspxFactory =
        javax.servlet.jsp.JspFactory.getDefaultFactory();

    private static java.util.Map<java.lang.String,java.lang.Long> _jspx_dependants;

```

02. 지시어

2. include 지시어

- include 지시어는 현재 JSP 파일에 다른 HTML이나 JSP 문서를 포함하기 위한 기능을 제공.
- include 지시어는 다음 절에서 살펴볼 include 액션과 비슷한 기능을 한다.

```
<%@ include file="포함할 파일_이름" %>
```

- 네이버와 같은 인터넷 포털사이트의 화면처럼 여러 정보의 조합으로 한 화면을 구성할 때 유용하게 사용됨.
- include 지시어를 사용하면 기능 혹은 화면을 모듈화할 수 있어 화면 구성이나 재활용이 용이하다.

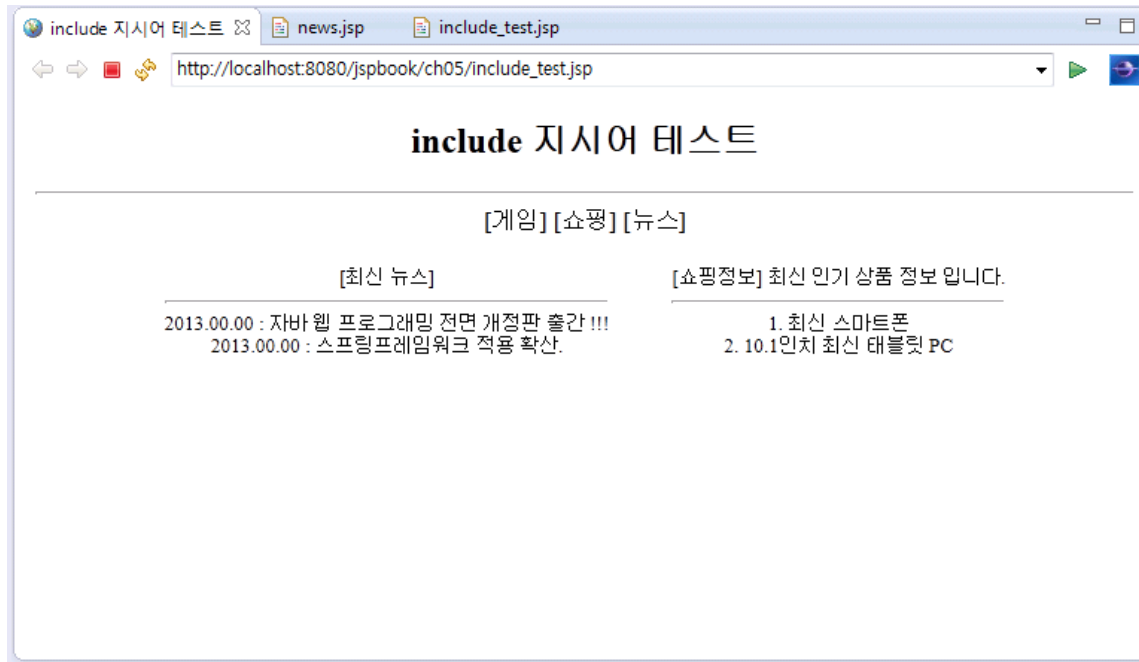


- ❶ 배너 화면 구성 : 광고 등 관련 사이트를 배너로 구성한다.
- ❷ 네이버 서비스 : 디렉터리 데이터베이스 연동 후 등록된 목록을 가져온다.
- ❸ 뉴스캐스트 : 뉴스 데이터베이스 연동 후 조회 수가 높은 최신 기사 제목들을 가져 온다.
- ❹ 쇼핑 : 쇼핑몰 데이터베이스 연동 후 인기 상품 이미지와 제품 정보를 가져 온다.

[그림 5-13] 네이버 화면 구성

02. 지시어

- [실습] 배너 화면 구성(include_test.jsp)
 - 교재 p.175 참고
- [실습] 메인 메뉴(menu.jsp), 뉴스캐스트(news.jsp), 쇼핑(shopping.jsp)
 - 교재 p.176 참고



[그림 5-14] 결과 확인

[예제 5-5] include_test.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <TITLE>ch05 : include 지시어 테스트 </TITLE></HEAD>
7 <BODY>
8 <div align="CENTER">
9 <H2> include 지시어 테스트</H2>
10 <HR>
11 <%@ include file="menu.jsp" %>
12 <P>
13 <table border=0 cellpadding=5 cellspacing=1>
14 <tr>
15 <td><font size=-1><%@include file="news.jsp" %></font></td>
16 <td width="20">&nbsp;</td>
17 <td><font size=-1><%@include file="shopping.jsp" %></font></td>
18 </tr>
19 </table>
20 </div>
21 </BODY>
22 </HTML>
```

포함되는 JSP 페이지

■ menu.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2
3 [게임] [쇼핑] [뉴스]
```

■ news.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 [최신 뉴스]<HR>
3 2013.00.00 : 자바 웹 프로그래밍 전면 개정판 출간 !!! <BR>
4 2013.00.00 : 스프링프레임워크 적용 확산.<BR>
```

■ shopping.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2
3 [쇼핑정보] 최신 인기 상품 정보 입니다.<HR>
4 1. 최신 스마트폰 <BR>
5 2. 10.1인치 최신 태블릿 PC <BR>
```


생성되는 HTML 출력 파일

```
include_test[1] - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>ch05 : include 지시어 테스트 </TITLE></HEAD>
<BODY>
<div align="CENTER">
<H2> include 지시어 테스트</H2>
<HR>

[게임] [쇼핑] [뉴스]
<P>
<table border=0 cellpadding=5 cellspacing=1>
  <tr>
    <td><font size=-1>
[최신 뉴스]<HR>
2013.00.00 : 자바 웹 프로그래밍 전면 개정판 출간 !!! <BR>
2013.00.00 : 스프링프레임워크 적용 확산.<BR>
    </font></td>

    <td width="20">&nbsp;</td>

    <td><font size=-1>
[쇼핑정보] 최신 인기 상품 정보 입니다.<HR>
1. 최신 스마트폰 <BR>
2. 10.1인치 최신 태블릿 PC <BR>
    </font></td>
  </tr>
</table>
</div>
</BODY>
</HTML>
```

02. 지시어

3. taglib 지시어

- JSP 기능을 확장하기 위해 만들어진 커스텀 태그 라이브러리를 사용하기 위한 지시어.
- 태그 라이브러리는 10장에서 자세히 살펴볼 것이고 여기서는 간단한 문법만 참조.

```
<%@ taglib uri="/META-INF/mytag.tld" prefix="mytag" %>
```

- 커스텀 태그는 공통으로 활용하거나 특정 기능을 HTML 태그 형태로 모듈화 하는 기술임
- 다음 예제에서는 GetInfo 라는 커스텀 태그를 사용하는 예로 실제 기능은 별도로 구현되어 있고 실행 결과는 user1 이라는 사용자의 정보를 DB에서 가져와 출력하는 형태가 될 수 있다.

```
01 <%@ page contentType="text/html; charset=UTF-8" %>
```

```
02 <%@ taglib uri="/WEB-INF/tlds/mytag.tld" prefix="mytag" %>
```

taglib 지시어

```
03 <HTML>
```

```
04 <BODY>
```

```
05 <mytag:GetInfo name="user1" />
```

커스텀 태그 사용

```
06 </BODY>
```

```
07 </HTML>
```

02. 지시어



저자 한마디

대규모 사이트에서의 분할 관리

include 지시어는 분명 편리한 페이지 분할 관리 기법을 제공하지만, 성능적인 측면에서는 그다지 권장할 만한 기법은 아니다. 또한 대규모 사이트의 구성에는 단순한 페이지 분할뿐만 아니라, 레이아웃 관리를 비롯한 고급 기능이 요구되므로 JSP의 include 지시어보다는 다음과 같은 별도의 전용 프레임워크를 사용하는 것이 더 적절한데, 스프링 프레임워크와 함께 사용하는 경우가 많다. 이들 프레임워크는 비교적 복잡한 설정과정이 필요하고 구조를 잘 설계해야 하므로 간단한 사이트 구성 등에는 include가 유용하게 사용될 수 있다.

- 타일즈(<http://tiles.apache.org>)
- 사이트 메시(<http://www.sitemesh.org>)

include 내부 실행 구조

[예제 5-5]의 실행결과 화면에서 [소스보기]를 하면 jsp 파일 4개가 모두 하나로 합쳐져 있다. 실제 java로 변환된 소스를 보더라도 include_test.jsp 이외의 파일은 자바 소스와 클래스가 생성되어 있지 않다. 즉 include는 각각의 파일을 포함한 뒤 하나의 컴파일된 클래스를 생성하므로 include만을 목적으로 사용할 파일이라면 소스에서 page 지시어 이외의 HTML 기본 태그 관련 부분들(<html>, <head>, <body> 등)은 생략할 수 있다.

URI와 URL의 차이점

URI는 'Uniform Resource Identifier'의 약자로, 필요로 하는 자원(파일 등)의 위치 정보를 기술하는데 사용한다. 일반적으로 알고 있는 URL과의 차이점은 프로토콜, 주소, 포트 번호를 제외한 경로만을 표시한다는 것이다.

- URL 예 <http://www.hanbitbook.com:8080/jsp/book/index.jsp>
- URI 예 [/jsp/book/index.jsp](#)



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음