

# 3장 JSP프로그래밍의 개요

컴퓨터정보과 김진숙

# 학습목표

- JSP페이지 개요
- JSP페이지 처리과정
- 웹 애플리케이션의 구조

# 1. JSP페이지 개요(1/2)

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>JSP 입문</title>
8   </head>
9
10  <body>
11    <%= "JSP 입문입니다." %>
12  </body>
13 </html>
```

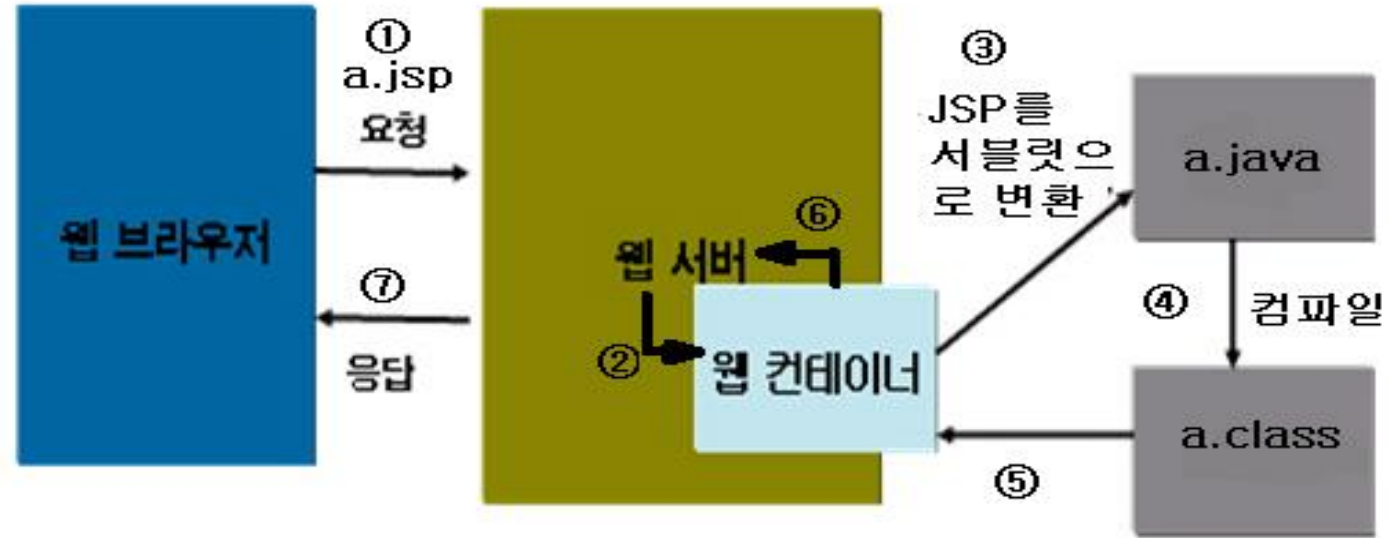
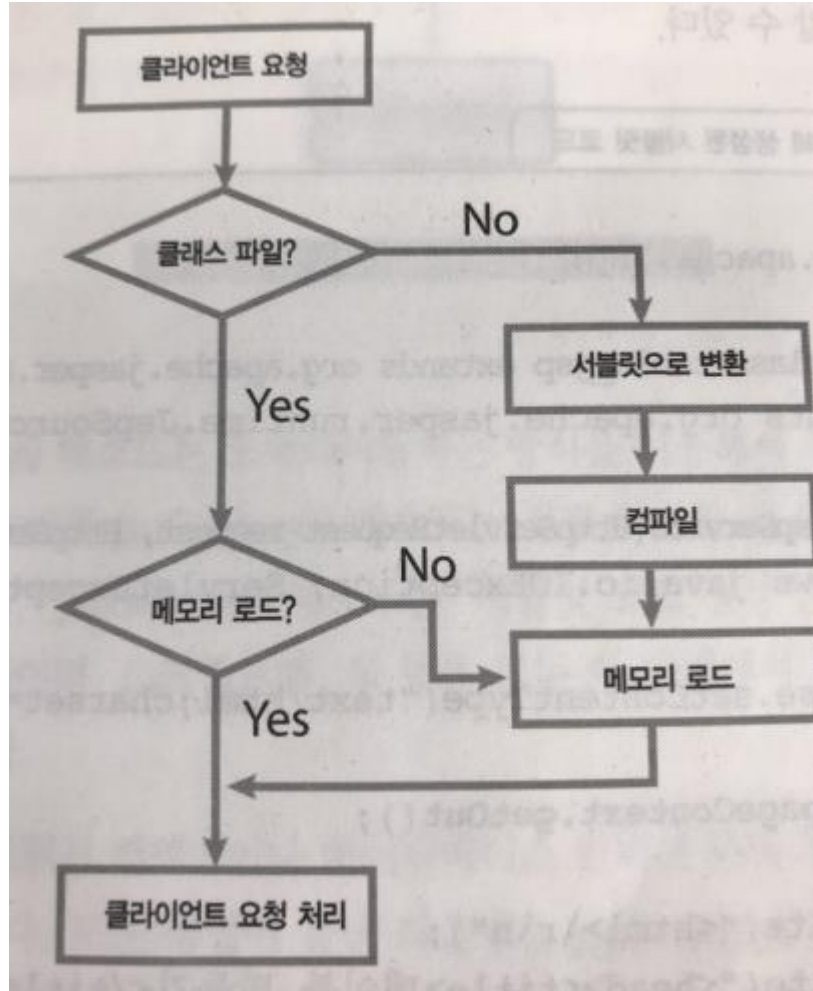
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>JSP 입문</title>
  </head>

  <body>
    JSP 입문입니다.
  </body>
</html>
```

# 1. JSP페이지 개요(2/2)

- JSP페이지 = **HTML** + **JSP**코드
- **HTML**부분 : 소스 코드의 정보가 모두 표시
  - 단순히 화면에 표시해야 하는 내용에 사용
- **JSP**페이지 : JSP코드에 기술한 코드는 화면표시 않됨
  - 중요 정보를 가지고 있는 코드는 JSP코드에 기술해서 표시되는 것 방지(**코드 보안** 기능)

## 2. JSP페이지의 처리과정(1/2)



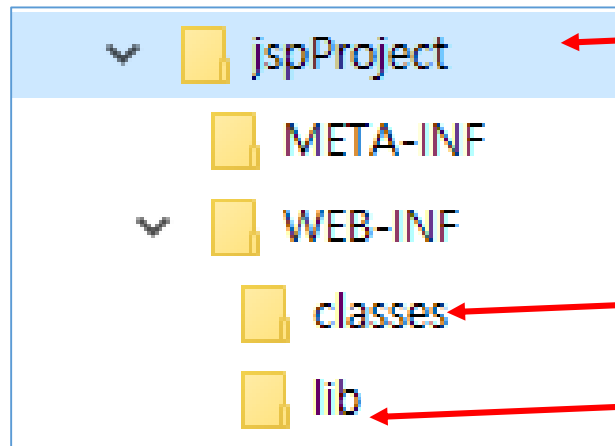
## 2. JSP페이지의 처리과정(2/2)

1. 웹 브라우저에서 `http://serverURL/xxx.jsp`과 같은 형태로 JSP페이지를 웹 서버에게 **요청**
2. 웹 서버는 JSP에 대한 요청을 **JSP 컨테이너(Jasper)**로 넘김
3. JSP 컨테이너는 서블릿(java파일 생성)으로 변환하는 파싱(parsing:구문분석 및 변환)의 과정을 실행 해 **서블릿 파일 생성**(xxx\_jsp.java)
  - 해당 JSP페이지가 처음 요청된 것일 경우 실행됨
  - 이전에 요청되었던 페이지는 다시 파싱할 필요가 없으므로 바로 5.단계로 감.
4. 서블릿은 **컴파일**(xxx\_jsp.class파일 생성)됨
  - JSP페이지가 최초로 호출된 때만 실시
5. 클래스 파일은 메모리에 로딩되어 **실행**됨
6. 실행결과는 **웹서버**에 넘겨짐
7. 웹서버는 웹 브라우저가 인식할 수 있는 **HTML** 로 결과를 **웹 브라우저**로 보냄

### 3. 웹 애플리케이션의 구조

- tomcat에서 웹 애플리케이션을 서비스 할 때 가상 경로
  - 웹 애플리케이션은 서비스하기 위한 프로그램
    - <http://127.0.0.1:8080/jspProject>
  - 웹상의 **가상적인 경로**로 실제 이들 웹 애플리케이션을 서비스하기 위한 파일은 로컬상에 웹 애플리케이션과 쌍을 이루는 폴더로 만들어짐
- 서버 컴퓨터의 실제 경로
  - [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\jspProject](#)

- 웹 애플리케이션 폴더 기본 구조

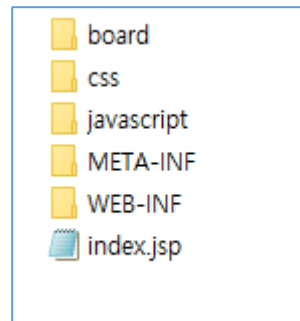
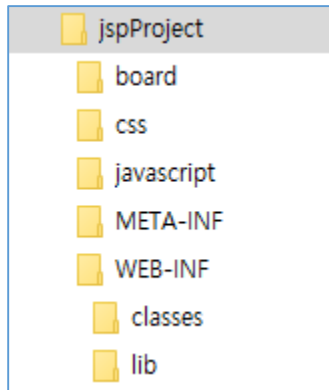


웹애플리케이션 / 웹프로젝트  
context Root 등

이 웹애플리케이션에서 사용하는 클래스 파일 모음

이 웹애플리케이션에서 사용하는 클래스 jar 파일 모음

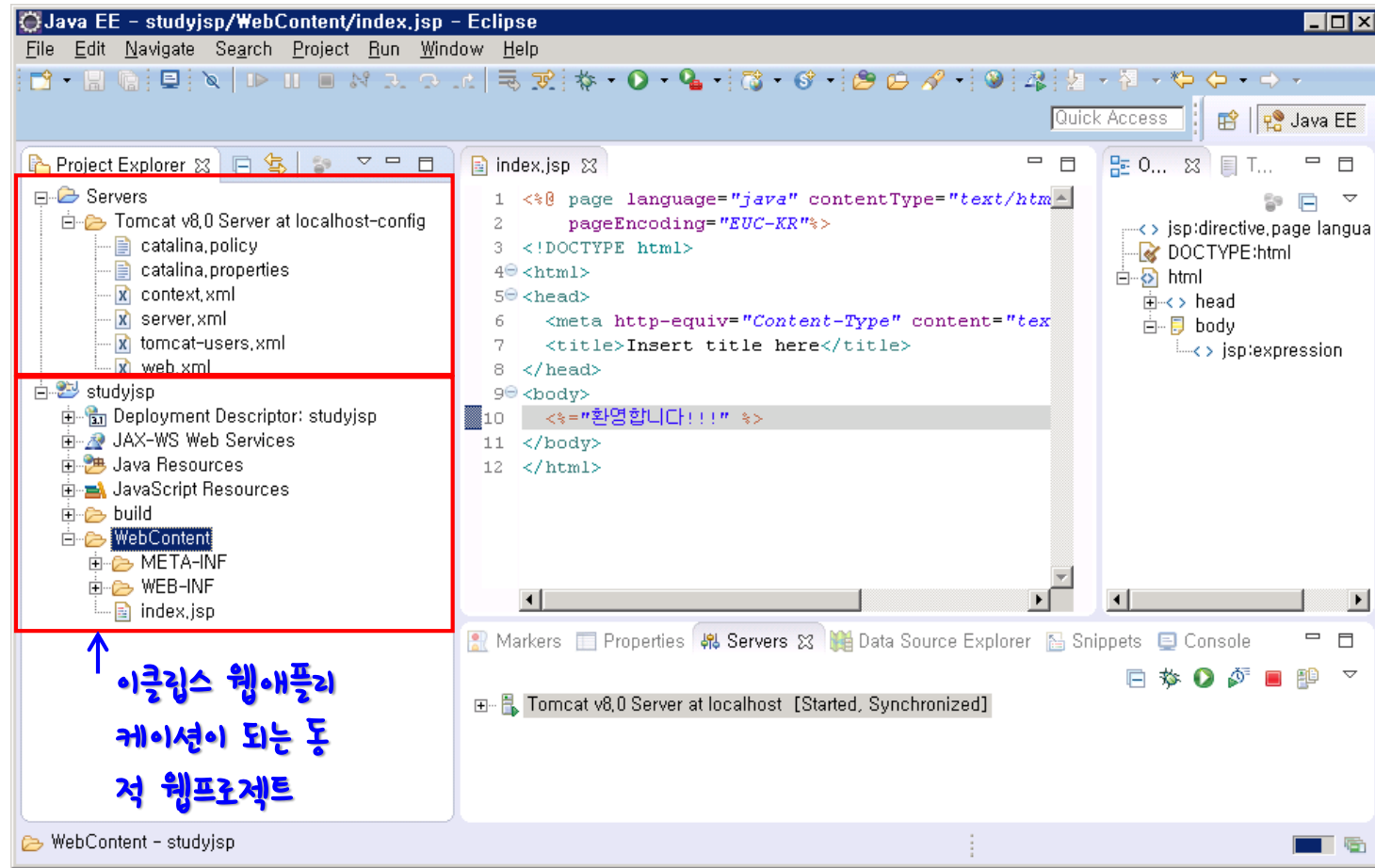
- 필요한 폴더와 파일들을 구성하여 웹 애플리케이션이 제작됨



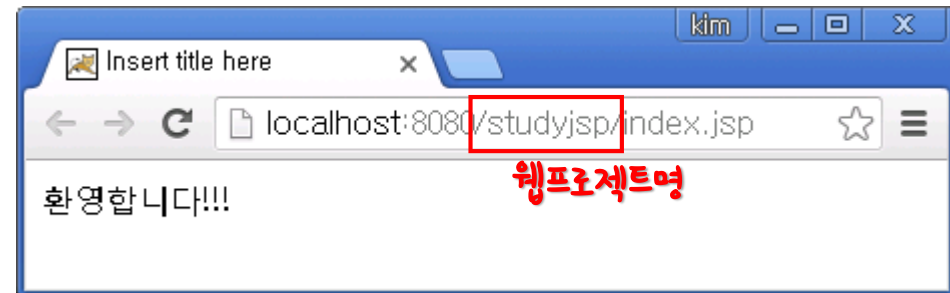


# 이클립스에서 웹 애플리케이션의 기본구조

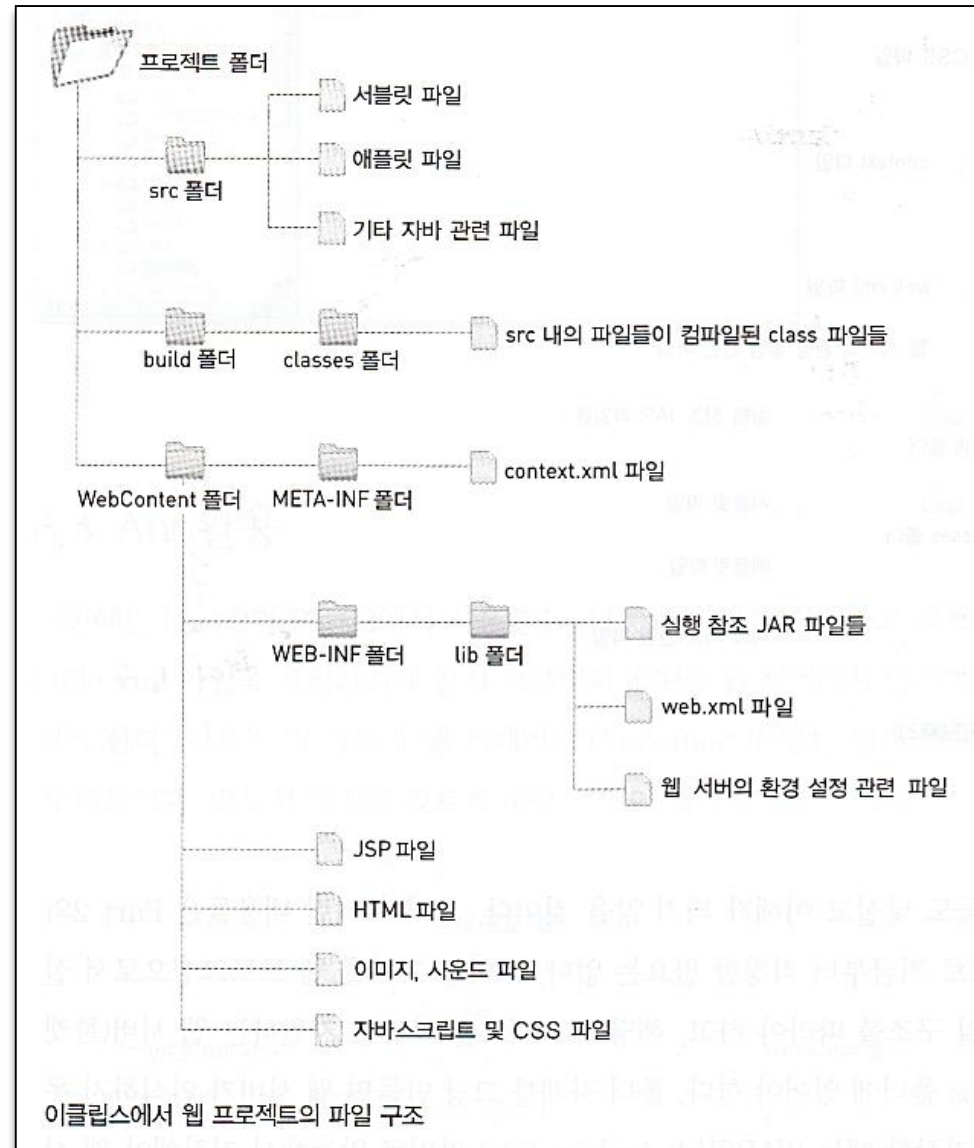
이클립스 가상환경  
웹서버 설정 파일



이클립스 웹애플리  
케이션이 되는 동  
적 웹프로젝트



# 이클립스에서 웹프로젝트의 파일 구조



### 3. 실제 웹서비스(웹브라우저에서 실행)

- studyjsp를 웹서비스 할 수 있도록 export실습
  1. .war( **W**eb **A**pplication a**R**chive) 파일로 내보내기
    - 위치는 C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps
  2. Tomcat 서버 구동하기
    - 이클립스에서 구동하는 서버 중지하기
    - C:\Program Files\Apache Software Foundation\Tomcat 9.0\startup.bat 실행하기
  3. 웹브라우저(chrome) 실행하기
    - <http://127.0.0.1:8080/jspProject/index.jsp> 실행

가상 환경인 이클립스에서 서버가 실행되면  
실제 환경인 톰캣 환경에서 서비스가 되지 않고  
실제 환경인 톰캣 환경에 서버가 구동 되면  
가상 환경인 이클립스에서 서버를 사용할 수 없다.

### 3. 실제 웹서비스(웹브라우저에서 실행)

- 서블릿으로 변환된 파일 확인

- 실제 서비스환경 :

- C:\Program Files\Apache Software Foundation\Tomcat 9.0\work\Catalina\localhost\studyjsp\org\apache\jsp

- 이클립스 가상환경 :

- C:\Users\jinsook\Desktop\김진숙

- \.metadata\plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\jspProject\org\apache\jsp

# 요약

- JSP페이지 개요
- JSP페이지 처리과정
- 웹 어플리케이션의 구조

# 4장 JSP 페이지의 디렉티브 (Directive, 지시자)

컴퓨터정보과 김진숙

# 학습목표

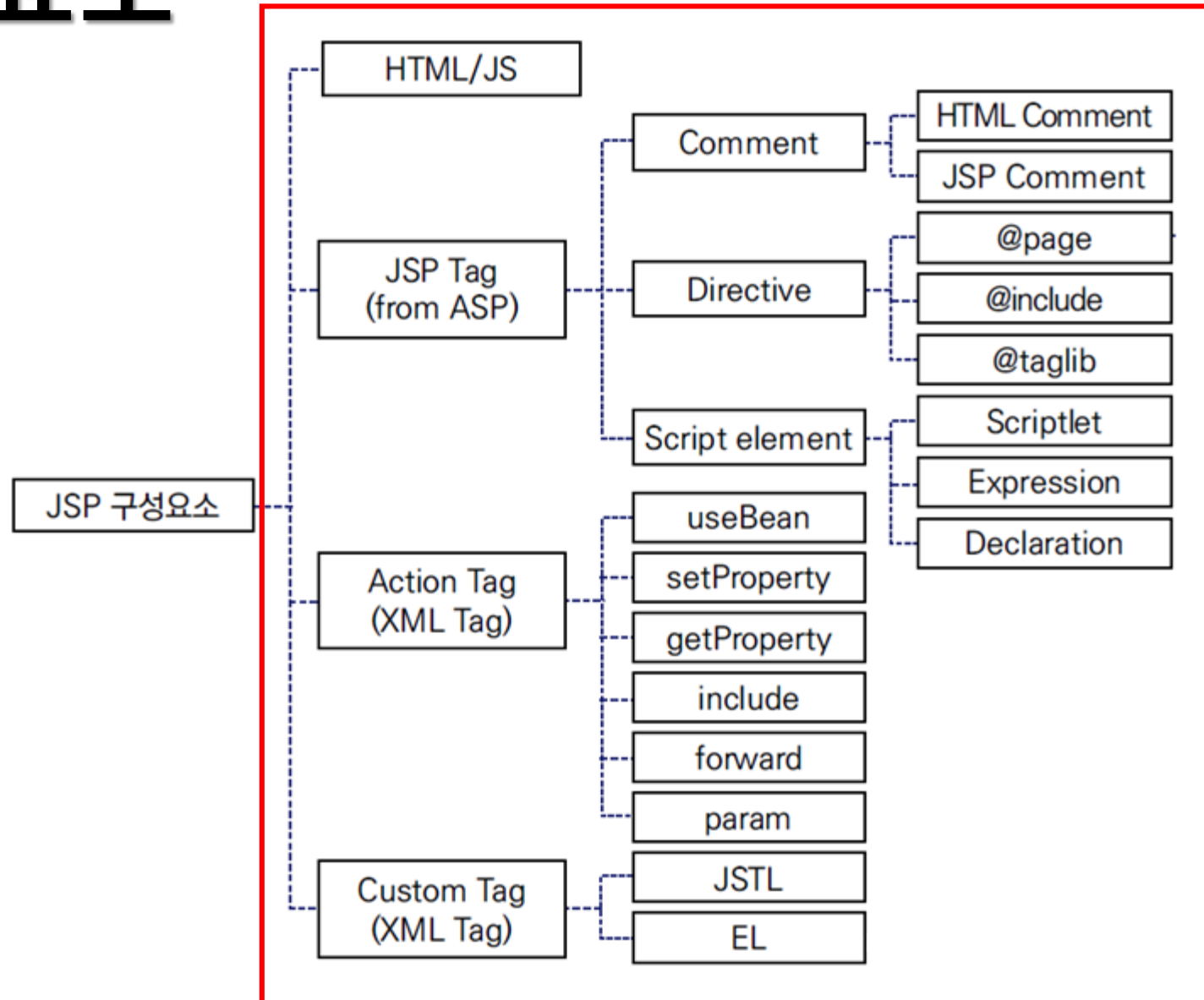
- 디렉티브(Directive)에 대해 학습
- page, include, taglib에 대해 자세히 학습

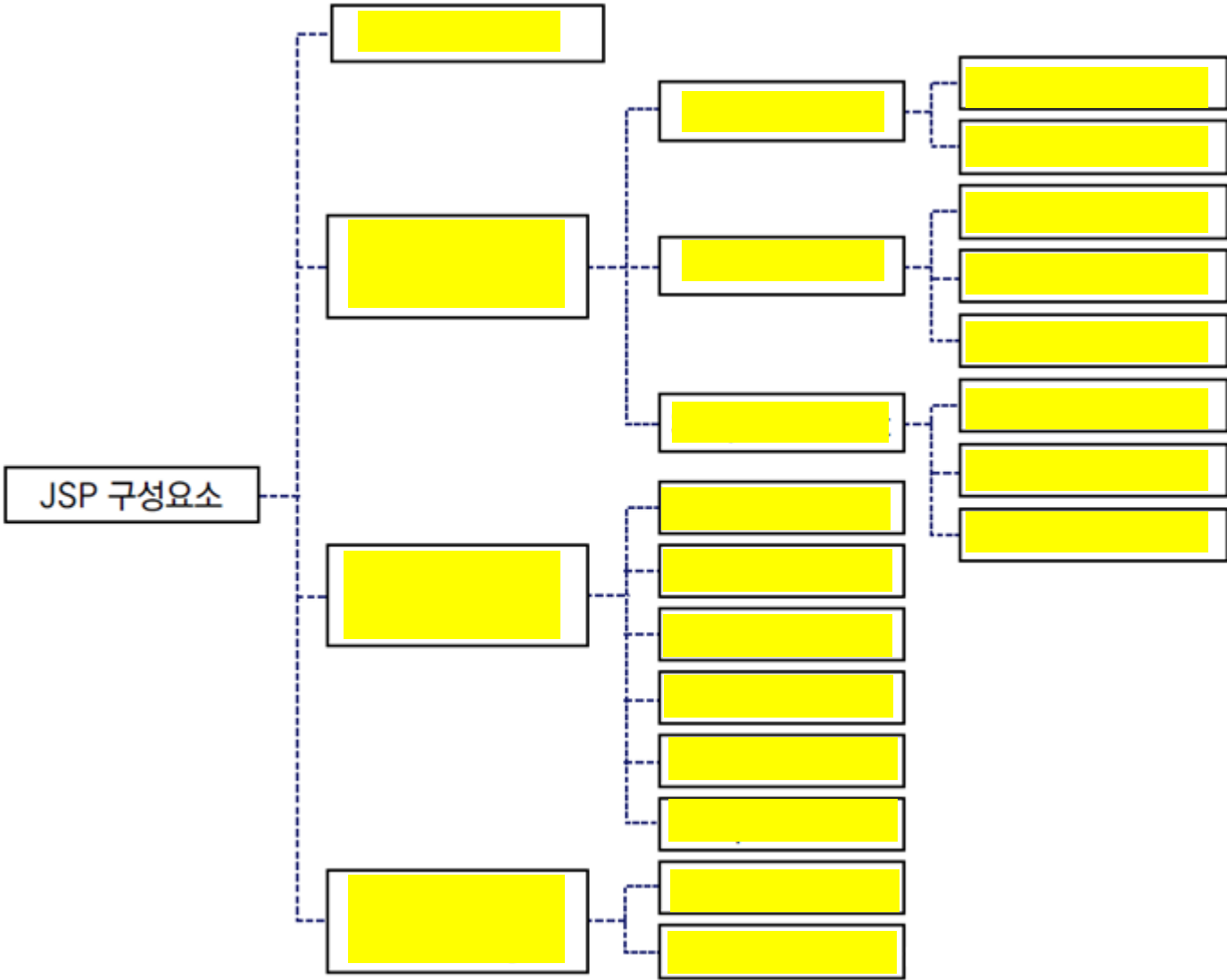
# 이장의 목차

- **page** 디렉티브(Directive)  
    <%@ page%>
- **include** 디렉티브(Directive)  
    <%@ include%>
- **taglib** 디렉티브 (Directive) – JSTL 사용 시 필요  
    <%@ taglib%>



# JSP 구성요소





# 지시자(Directive)

디렉티브	설명	사용 예
page	해당 JSP 페이지 전반적인 환경을 설정할 내용을 지정한다.	<pre>&lt;%@ page contentType="text/html; charset=utf-8" %&gt; &lt;%@ page import="java.util.*" %&gt;</pre>
include	현재 페이지에 다른 파일의 내용을 삽입할 때 사용된다.	<pre>&lt;%@ include file="test.jsp"%&gt;</pre>
taglib	태그 라이브러리에서 태그를 꺼내와 사용할 수 있는 기능을 제공한다.	<pre>&lt;%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%&gt;</pre>

# 1. page 지시자(Directive)(1/4)

- JSP 페이지에 대한 설정 정보를 속성을 사용해서 정의
  - 생성되는 문서의 타입, import할 클래스, 세션 및 버퍼의 사용여부, 버퍼의 크기 등
  - 예 : `<%@ page contentType="text/html;charset=utf-8"%>`
- page 디렉티브의 속성은 **101쪽**을 볼 것

# 1. page 지시자(Directive)(2/4)

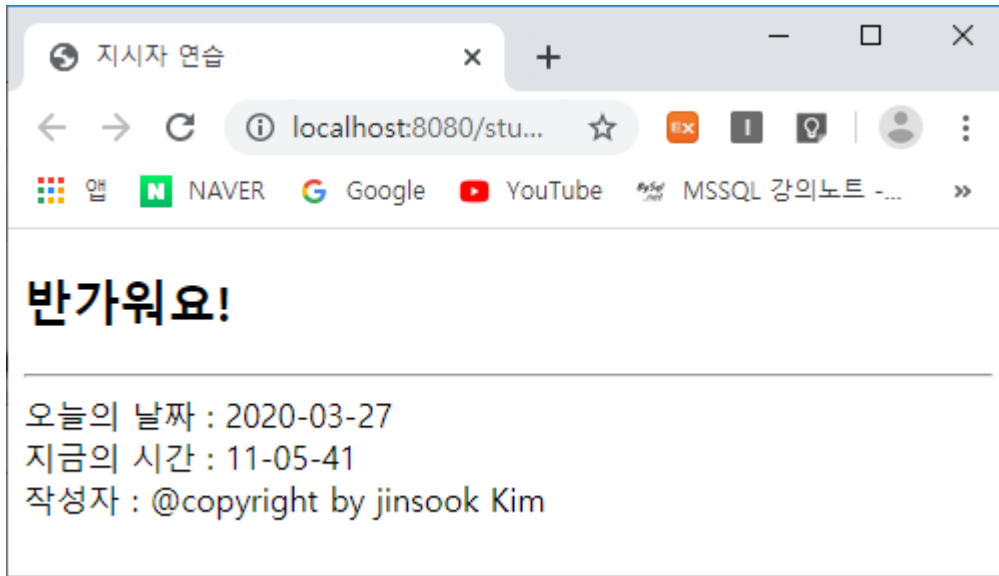
속성	설명
<b>info</b>	<ul style="list-style-type: none"><li>- 페이지를 설명해 주는 문자열(페이지 제목)</li></ul> <code>&lt;%@ page info="Business Management System" %&gt;</code>
<b>language</b>	<ul style="list-style-type: none"><li>- JSP 페이지의 스크립트 요소에서 사용할 언어를 지정하는 속성</li></ul> <code>&lt;%@ page language="java" %&gt;</code> 기본값 : java
<b>contentType</b>	<ul style="list-style-type: none"><li>- 서버에서 웹브라우저가 받을 페이지의 인코딩 방식 설정</li><li>- HTTP 헤더에 그 내용이 들어감</li></ul> <code>&lt;%@ page contentType="text/html" %&gt;</code> 기본값 : text/html
<b>pageEncoding</b>	<ul style="list-style-type: none"><li>- JSP 페이지 소스코드의 문자(character)의 인코딩을 지정할 때 사용</li><li>- JSP 페이지에 기록된 소스 코드 자체의 인코딩 방식 설정</li></ul>
<b>import</b>	<ul style="list-style-type: none"><li>- 다른 패키지에 있는 클래스를 가져다 쓸 때 사용</li><li>- import 속성은 page 지시자 중에 유일하게 <b>중복 사용 가능</b></li></ul>

# 1. page 지시자(Directive)(3/4)

속성	설명
<b>isThreadSafe</b>	<ul style="list-style-type: none"><li>- 하나의 JSP 페이지가 동시에 여러 브라우저의 요청을 처리할 수 있는지 여부를 설정</li><li>- 멀티쓰레드 허용여부 설정</li></ul>
<b>autoFlush</b>	<ul style="list-style-type: none"><li>- JSP 페이지 내용이 브라우저에 출력이 되기 전 버퍼에 저장되어 있는 내용 처리방법을 결정 하는 속성</li><li>- 기본값은 true</li></ul>
<b>buffer</b>	<ul style="list-style-type: none"><li>- JSP 페이지의 출력 크기를 킬로바이트(KB) 단위로 지정하는 속성</li><li>- 기본값은 "8KB"(데이터 전송 단위)</li></ul>
<b>session</b>	<ul style="list-style-type: none"><li>- HttpSession을 사용할지 여부를 지정하는 속성 (true가 기본값)</li></ul>
<b>trimDirectiveWhite spaces</b>	<ul style="list-style-type: none"><li>- 결과 페이지(HTML)에서 빈 공간(JSP가 차지했던)을 없애는 속성</li></ul>

# 실습 - pageDirective

- 사용할 속성
  - info
  - import
  - trimDirectiveWhitespaces



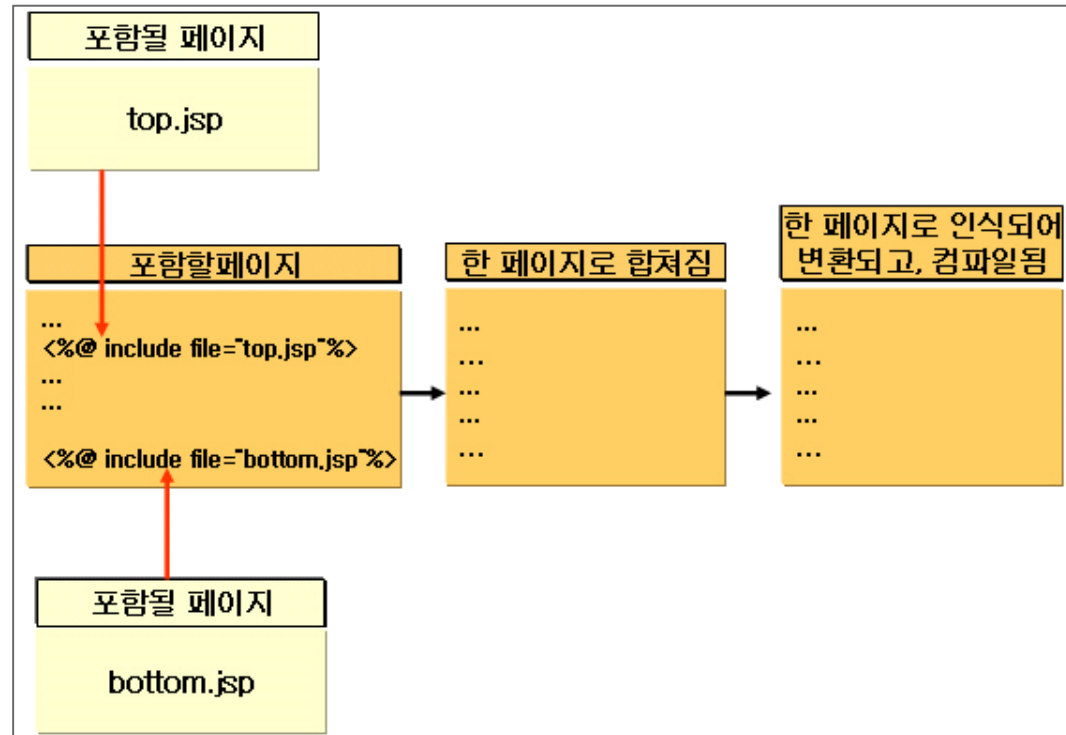
## 2. include 지시자(Directive) (1/3)

- 한 페이지 내에 다른 페이지를 삽입
  - 공통적으로 포함하는 내용을 가진 파일을 해당 JSP 페이지 내에 삽입하도록 하는 것
- include 디렉티브 사용법
  - `<%@ include file="포함될 파일의 URL"%>`
- include 디렉티브는 주로 **조각 코드**를 삽입할 때 사용



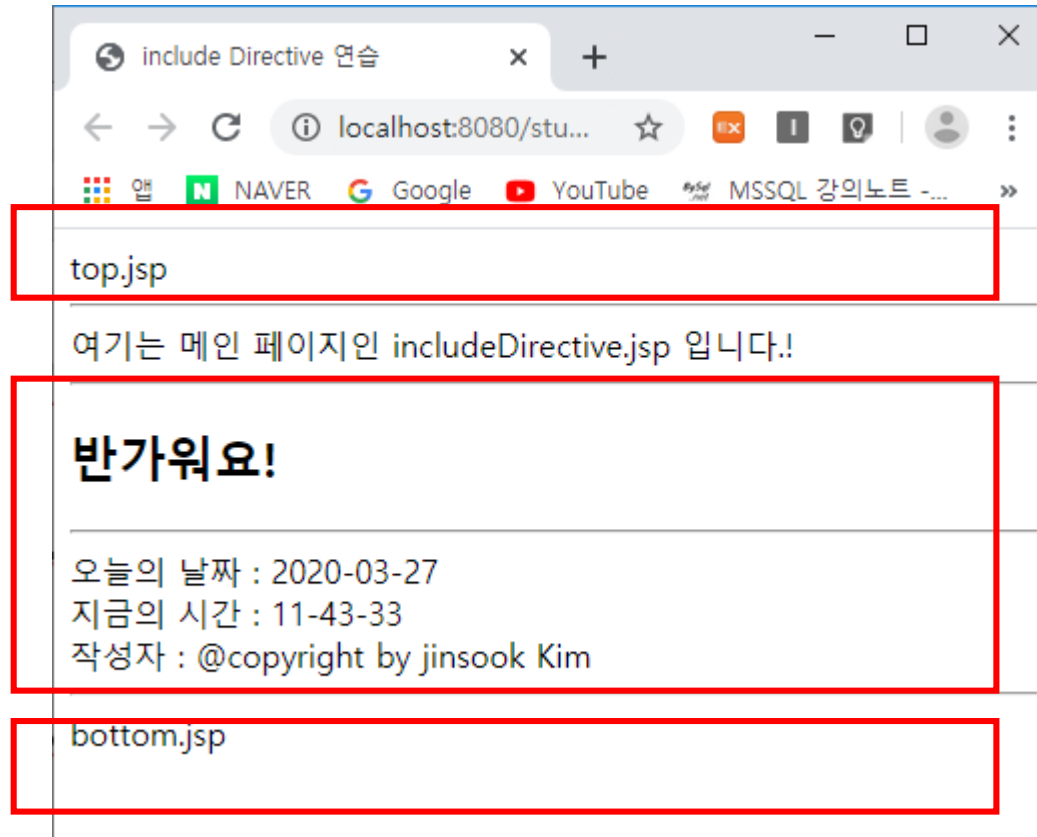
## 2. include 지시자(Directive) (2/3)

- include 디렉티브의 처리 과정은 **정적(static)**
  - include 지시자를 사용한 JSP 페이지가 컴파일 되는 과정에서 include 되는 JSP페이지의 소스 내용을 그대로 포함해서 컴파일됨
  - 복사 & 붙여넣기 방식으로 두 개의 파일이 하나의 파일로 구성된 후 같이 변환되고 컴파일



# 실습 - includeDirective

- includeDirective.jsp 에서 아래 3개의 파일을 include 하여 실행
  - top.jsp
  - pageDirective.jsp
  - bottom.jsp



# 3.스크립트릿(Scriptlet)

- 가장 일반적인 스크립트 요소
- 프로그래밍의 로직 기술
- JSP 페이지가 Servlet으로 변환되고 이 페이지가 호출 될 때 **\_jspService** 메소드 안에 선언됨
- 스크립트릿(Scriptlet)에서 선언된 변수는 **지역변수**로 선언
- 스크립트릿의 문법  
<% 문장 %>

```
<%  
    String id = "jinsook";  
    String pass="";  
%>
```

- **멤버변수** : 전역변수와 유사. 초기값을 기술하지 않으면 기본 값으로 초기화됨. 숫자는 0, 문자나 레퍼런스 타입은 Null로 초기화
- **지역변수** : 메소드 안에 선언된 변수로 선언된 메소드 내에서만 유효함. 초기화가 자동으로 일어나지 않으므로 반드시 초기화 할 것

## 4. 표현식(Expression)

- 웹브라우저 화면 출력 부분
- 변수값, 메소드 결과값 출력
  - 스크립트릿 내에서 표현식 사용 불가능
  - 스크립트릿 내에서 출력할 부분은 내장객체인 out객체의 print()또는 println() 메소드를 사용
- 표현식내에서는 **세미콜론(;) 생략**
  - JSP페이지가 서블릿으로 변환될 때 표현식 부분은 out.write(); 메소드로 변환되어 자동적으로 세미콜론이 붙여지기 때문
- 표현식의 일반적인 문법
  - <%=문장%>
  - <%=name[i]%>

# 5. 주석(Comment)

- 프로그램의 실행에 영향을 미치지 않는지만 프로그램의 이해 및 소스 코드를 분석 위해서는 꼭 필요한 것이니 **반드시** 기술할 것
- 좋은 프로그래밍 스타일에 가장 중요한 것이 주석 달기

구분	표기 방식	설명
HTML 주석	<!-- ... -->	<ul style="list-style-type: none"><li>• &lt;!-- html 주석입니다. &amp; <b>&lt;%=str%&gt;</b> --&gt;</li><li>• HTML주석문의 경우 화면에 표시는 되지 않으나 &lt;%=str%&gt;와 같은 JSP실행코드들이 실행된다.</li></ul>
JSP 주석	<%-- --%>	<ul style="list-style-type: none"><li>• JSP 페이지에서만 사용</li></ul>
Java 주석	// : 한 줄 /* ... */ : 여러 줄	<ul style="list-style-type: none"><li>• 자바 주석은 실제로 가장 많이 사용되는 주석으로, <b>스크립트릿</b>이나 <b>선언문</b>에서 사용되는 주석으로, 자바와 주석 처리 방법이 같다.</li><li>• JSP주석이 서블릿으로 변환 시 자바주석으로 변환됨</li></ul>

# 톰캣기반에서의 한글처리

6장 3의 내용(198쪽 내용)

## 1. 서버 → 웹 브라우저

- 서버로부터 클라이언트로 응답되는 페이지의 화면 출력 시 한글처리
- `<%@ page contentType= " text/html; charset=utf-8 " %>`
- 서버페이지에 반드시 명시해야 함

# 톰캣기반에서의 한글처리

## 2. 웹 브라우저 → 서버(Post방식)

- 웹브라우저에서 POST방식으로 데이터가 넘어오는 파라미터 값에 한글이 있는 경우
- `<% request.setCharacterEncoding("utf-8") %>`
- `<% request.setCharacterEncoding("utf-8") %>`부터 파라미터를 넘겨받는 페이지에는 반드시 위 코드를 작성해야함

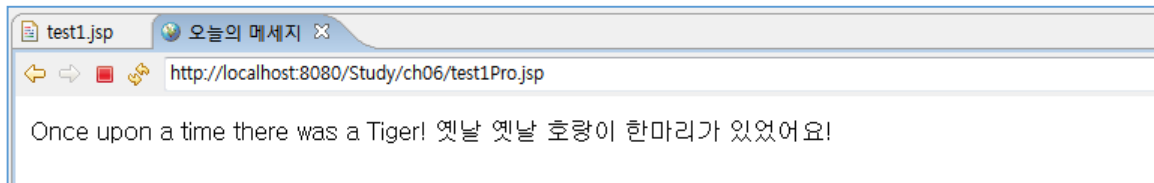
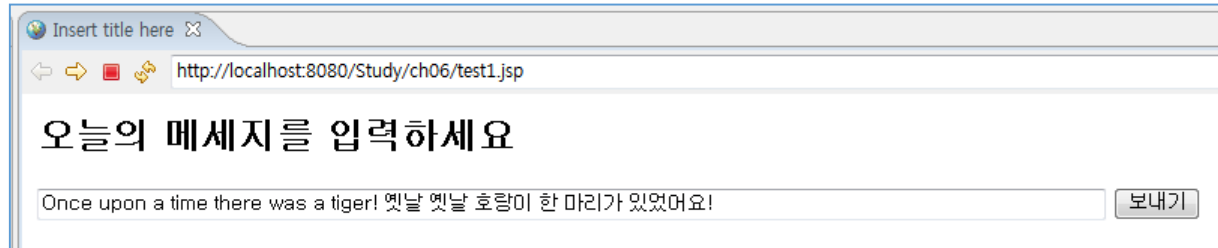
## 3. 웹 브라우저 → 서버(Get방식)

- 별 다른 처리가 필요하지 않음
- 지난 버전에서는 server.xml에 다음과 같이 추가해주어야 했으나 tomcat9 은 처리 필요하지 않음

```
<Connector connectionTimeout="20000" port="8080"
  protocol="HTTP/1.1" redirectPort="8443"
  URIEncoding="utf-8"/>
```

# 실습

- 폼에서 오늘의 메시지를 입력하면 처리하도록 하는 JSP를 작성하시오





# 실습

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <!DOCTYPE html>
5 <html>
6 <head>
7   <meta charset=UTF-8>
8   <title>오늘의 메시지</title>
9 </head>
10 <body>
11   <h1>오늘의 메시지를 입력하세요!</h1><hr/>
12   <form method="post" action="p2.jsp">
13     <input type="text" name="msg" size="80">
14     <input type="submit" value="보내기">
15   </form>
16
17 </body>
18 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%
4   request.setCharacterEncoding("utf-8");
5
6   String msg = request.getParameter("msg");
7   out.println(msg);
8 %>
```