

Chapter 04

JSP의 기본 문법

JSP의 구성 요소

JSP의 구성 요소

■ JSP의 구성 요소

- 디렉티브(directive) (지시어)

- ▶ JSP 파일의 속성을 기술하는 JSP 문법
- ▶ JSP 컨테이너에게 해당 페이지를 어떻게 처리해야 하는지 전달하기 위한 내용을 담고 있음
- ▶ 지시어는 크게 page, include, taglib으로 구성되며, 각 지시어에서 다루는 속성이 다름

- 스크립트 요소 (script elements)

- ▶ JSP 스크립트를 구성하는 요소
- ▶ 주석문, 스크립트릿(scriptlet), 표현식(expression), 선언(declaration)

- 내장 객체 (Implicit Object)

- ▶ JSP가 기본적으로 제공하는 객체로 인스턴스 생성 없이 사용 가능
- ▶ request, response, out, page, session, application 객체 등

JSP의 구성 요소

- 액션 태그(action tag)
 - JSP 문서 사이의 제어의 이동, 자바 빈즈 컴포넌트의 사용을 위한 태그
- 표현 언어(expression language)
 - 자바 코드가 들어가는 표현식을 편리하게 사용하기 위해 JSP2.0부터 도입된 데이터 출력 기능
- 커스텀 태그(custom tag)
 - 개발자가 특정 기능을 수행하는 태그를 직접 생성하는 방법
 - JSTL(JSP Standard Tag Library)은 대표적인 커스텀 태그임

지시어 [directives]

지시어의 개요

■ 지시어(directives)

- JSP 파일의 속성을 기술하는 JSP 문법

- ▶ 해당 페이지에 대한 정보와 처리하기 위한 방법을 JSP 컨테이너에 전달하는 내용을 담고 있음

- 지시어의 종류

지시어	설명
페이지(page)	<ul style="list-style-type: none">- 현재 JSP 페이지를 컨테이너에서 처리하는 데 필요한 각종 속성을 기술- JSP 페이지의 전반적인 정보를 설정할 때 사용
인클루드 (include)	<ul style="list-style-type: none">- 현재 JSP 파일에 다른 HTML이나 JSP 문서를 포함하기 위한 기능을 제공- 공통으로 사용할 JSP 문서를 다른 JSP 문서에 삽입할 때 사용
태그라이브 (taglib)	<ul style="list-style-type: none">- 개발자가 custom tag를 사용하거나 framework에서 제공하는 태그를 이용할 때 사용

지시어의 개요

■ 지시어의 형식

```
<%@지시어 속성1="값1" 속성2="값2" ... %>
```

- 지시어를 구성하는 모든 내용은 <%@ 기호와 %> 기호 사이에 존재
 - 시작 기호인 <%@나 종료 기호인 %> 내부에는 공백이 있어서는 안됨
 - page directive일 경우 '<%@page ... '로 표현
- 각 디렉티브 태그마다 고유의 속성(property)과 값(value)을 가짐
 - 대소문자는 구분하므로 유의해야 함

page directive

페이지 지시어의 개요

■ 페이지 지시어

```
<%@ page 속성1="값1" 속성2="값2" ... %>
```

- 현재 JSP 페이지를 컨테이너에서 처리하는 데 필요한 각종 속성을 기술하는 부분
 - ▶ JSP 페이지 맨 앞에 위치함
- 이클립스 개발 도구를 이용해 jsp 파일을 생성하는 경우 기본적인 page 지시어는 자동 생성됨
 - ▶ 필요에 따라 속성을 추가해 사용
- 속성이 많이 사용되는 경우 여러 줄에 나누어 작성할 수 있음
 - ▶ [예]

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

페이지 지시어의 개요

■ 페이지 지시어의 속성

속성값	기본값	설명
info	없음	페이지를 설명해 주는 문자열을 지정
language	"java"	JSP 페이지에서 사용할 언어를 지정
contentType	"text/html"	JSP 페이지 출력 형식을 지정
pageEncoding	"ISO-8859-1"	JSP 페이지에서 사용하는 문자열 인코딩을 지정
import	없음	JSP 페이지에서 다른 패키지의 클래스를 import 할 때 지정
session	"true"	JSP 페이지에서 세션 객체의 사용 여부를 지정
buffer	"8kb"	JSP 페이지 출력 시 사용할 버퍼 크기를 지정
autoFlush	"true"	JSP 페이지의 내용이 출력되기 전 버퍼가 다 채워질 경우 동작을 지정
errorPage	"false"	JSP 페이지 처리 도중 예외가 발생할 경우 예외 처리 담당 페이지를 지정
isErrorPage	"false"	현재 JSP 페이지가 예외 처리를 수행하는 페이지인지를 지정
isELIgnored	"true"	JSP 2.0 버전에서 추가된 기능으로 EL 사용 유무를 지정

페이지 지시어의 속성

■ info 속성

```
<%@page info="문자열"%>
```

- JSP 문서에 대한 설명이나 버전, 작성자와 작성일과 같은 문자열 형태의 정보를 지정
 - 생략 가능

■ language 속성

```
<%@page language="개발언어"%>
```

- JSP 문서에서 사용할 언어를 지정
 - 현재 JSP가 지원하는 언어는 Java 밖에 없기 때문에 생략이 가능 (기본 값은 Java임)
 - 향후 JSP 언어가 다른 언어를 지원하도록 확장될 가능성을 염두에 두고 만든 것

페이지 지시어의 속성

■ contentType 속성

```
<%@page contentType="MIME type;charset=Encoding" %>
```

- 현재 JSP 문서의 MIME(Multipurpose Internet Mail Extension)을 지정
 - 문서의 처리 결과를 브라우저에 출력할 때 사용할 문자 인코딩을 지정
- 사용 예

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

- 브라우저에 전달되는 콘텐츠의 형태가 html 문서임을 지정
- tomcat에 의한 처리 결과를 브라우저에 출력할 때 사용할 문자 인코딩이 UTF-8임을 지정
- UTF-8을 지정하지 않으면 contentType 속성의 기본 값인 ISO-8859-1로 인식됨
 - 한글이 정상적으로 출력되지 않음

페이지 지시어의 속성

■ pageEncoding 속성

```
<%@page pageEncoding="인코딩 방식" %>
```

- tomcat 컨테이너에 전달할 인코딩 방식을 설정
- JSP2.0 스펙에 추가된 속성으로 이전 버전의 컨테이너의 경우 사용할 수 없음
- JVM과 JSP 사이에서의 문자 인코딩의 처리 과정
 - ▶ Java는 기본적으로 유니코드를 사용
 - ▶ JSP 문서가 서블릿으로 변환될 때 pageEncoding 속성에 지정된 인코딩을 유니코드로 변환
 - ▶ 브라우저에 출력될 때 변환된 유니코드는 contentType 속성으로 지정된 인코딩으로 변환

페이지 지시어의 속성

■ contentType과 pageEncoding의 처리

- 컨테이너는 먼저 pageEncoding에 정의된 인코딩을 찾음
 - 인코딩이 존재할 경우 해당 인코딩을 사용
 - 인코딩이 존재하지 않을 경우(pageEncoding 속성이 생략된 경우), contentType 속성을 찾음
 - 따라서, contentType 속성이 지정된 경우 pageEncoding은 생략 가능
 - 통상적으로는 함께 사용되고 있음

• [예]

```
<%@ page contentType="text/html;chaeset=euc-kr" pageEncoding="euc-kr" %>
```

- 컨테이너는 JSP를 읽어 들일 때 euc-kr 인코딩을 사용 (euc_kr → 유니코드)
- JSP 문서에 처리결과를 출력할 때 euc-kr 인코딩을 사용 (유니코드 → euc-kr)

- 두 인코딩은 일치시키는 것이 좋음
 - 가급적 utf-8을 사용하는 것이 좋음

페이지 지시어의 속성

- JSP 문서에서의 한글 처리

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 지정된 인코딩을 찾지 못할 경우 ISO8859-1을 적용

페이지 지시어의 속성

■ import 속성

```
<%@page import="java 클래스|패키지"%>
```

- JSP 프로그램 작성에 필요한 Java 패키지나 클래스를 포함시키는데 사용

- ▶ Java에서 import 구문과 동일한 역할을 수행

- 하나 이상의 클래스를 import하는 방법

- ▶ 콤마(,) 를 사용해 구분하는 방법

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

- ▶ 여러 개의 페이지 지시어를 사용하는 방법

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

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


페이지 지시어의 속성

■ session 속성

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

- 세션은 웹 브라우저와 웹 서버가 지속적인 연결을 위해 필요한 정보를 임시로 저장해두는 방법
 - 주로 웹 사이트에 로그인하거나 쇼핑몰에서 장바구니 등을 구현할 때 사용
- JSP는 기본적으로 세션을 사용하므로 기본 값은 true임
 - 일부러 사용을 제한할 경우에만 값을 false로 지정
 - 일반적으로 true를 사용
- 세션과 관련한 자세한 내용은 차후 자세히 학습

페이지 지시어의 속성

■ buffer 속성

```
<%@page buffer="버퍼 크기"%>
```

- 버퍼의 크기를 변경하는 기능 수행
- JSP는 out 객체를 사용해 브라우저로 출력할 내용을 전송
 - out 객체는 전송의 효율을 높이기 위해 버퍼를 사용함
 - buffer 속성을 사용하면 out 객체의 버퍼 크기를 변경할 수 있음
- 기본값은 8KB 임
 - 동적으로 많은 내용이 포함될 경우 버퍼 크기 조정이 필요하나 일반적으로는 변경하지 않음
- 버퍼에 대한 자세한 내용은 out 내장객체에서 다룸

페이지 지시어의 속성

■ autoFlush 속성

```
<%@page autoflush="none | true | false" %>
```

• flush란

- ▶ out 객체의 버퍼에 있는 내용을 브라우저로 전송하는 것(브라우저에 출력하는 것)

• autoflush란

- ▶ 버퍼가 가득 찰 경우, 버퍼에 있는 내용을 자동으로 브라우저에 전송하는 것을 의미
- ▶ 기본 값은 true 임 (일반적으로 버퍼를 사용 함)

속성 값	설명
none일 경우	- 버퍼를 사용하지 않음
true일 경우	- 버퍼가 가득 차면 버퍼 내의 모든 정보를 클라이언트로 전송 - 버퍼는 Tomcat으로부터 계속 정보를 받아들임
false일 경우	- 버퍼가 가득 차면 IOException을 발생시키고 수행 중이던 JSP 문서의 실행을 중단 - 프로그래머는 버퍼가 가득 차기 전에 주기적으로 flush시켜야 함

페이지 지시어의 속성

■ **errorPage**속성, **isErrorPage** 속성

- **errorPage** 속성

```
<%@page errorPage="오류처리문서" %>
```

- ▶ 현재 JSP 문서에서 예외가 발생한 경우, 예외를 처리하는 문서를 지정하는 역할을 수행
- ▶ 수행 중인 JSP 문서에서 예외가 발생한 경우, 인자로 지정한 문서가 처리하도록 지정

- **isErrorPage** 속성

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

- ▶ 현재의 JSP 문서가 일반 문서인지 아니면 예외를 처리하는 문서인지를 지정 (기본 값은 false)
- ▶ 현재의 문서가 예외를 처리하는 문서임을 지정

- 예외처리 부분에서 자세히 학습함

페이지 지시어의 속성

- `errorPage`와 `isErrorPage` 속성의 예

- ▶ 오류가 발생할 경우 일반적인 출력 내용

```
HTTP Status 500 - java.lang.NumberFormatException: For input string: "test"

Type Exception report
Message java.lang.NumberFormatException: For input string: "test"
Description The server encountered an internal error that prevented it from fulfilling this request.
Exception
org.apache.jasper.JasperException: java.lang.NumberFormatException: For input string: "test"
    org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:
    org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:470)
    org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
    org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
    jspbook.ch13.EncFilter.doFilter(EncFilter.java:28)
```

- ▶ 에러를 처리하는 속성을 사용하는 경우

```
처리중 문제가 발생했습니다

관리자에게 문의해 주세요..
빠른시일내 복구하겠습니다.

java.lang.NumberFormatException: For input string: "test"
```

페이지 지시어의 속성

- `errorPage`와 `isErrorPage` 속성의 예

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

오류가 발생될 경우 오류를 처리하는 페이지를 지정

errorProc.jsp

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

현재 페이지가 오류를 처리하는 페이지라는 것을 지정

페이지 지시어의 속성

■ isThreadSafe 속성

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

- 기본적으로 서블릿은 스레드로 동작하기 때문에 스레드로 인한 동기화 문제를 해결하기 위한 속성
 - ▶ 기본값은 true로, 일반적으로 false로 설정하는 경우는 거의 없음

include directive

인클루드 지시어

■ 인클루드 지시어

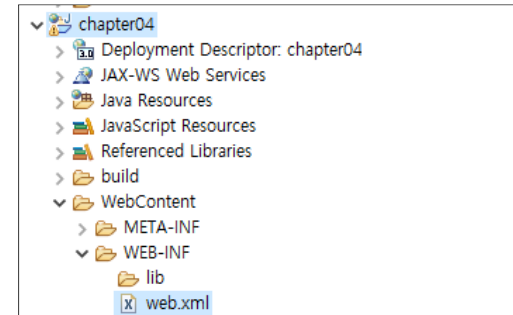
```
<%@include file="포함시킬 파일" %>
```

- 현재 JSP 파일에 다른 HTML이나 JSP 문서를 포함하기 위한 기능을 제공
 - file 속성을 사용해 포함시킬 문서를 지정
 - 기능이나 화면을 모듈화할 수 있어 화면 구성이나 재활용이 용이함
- <jsp:include> 액션 태그와 유사한 동작을 수행함
 - 차후 액션 태그에서 자세히 학습

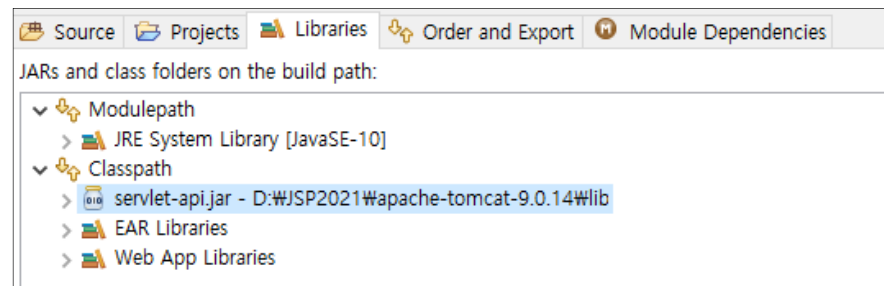
인클루드 지시어

■ 프로젝트 생성

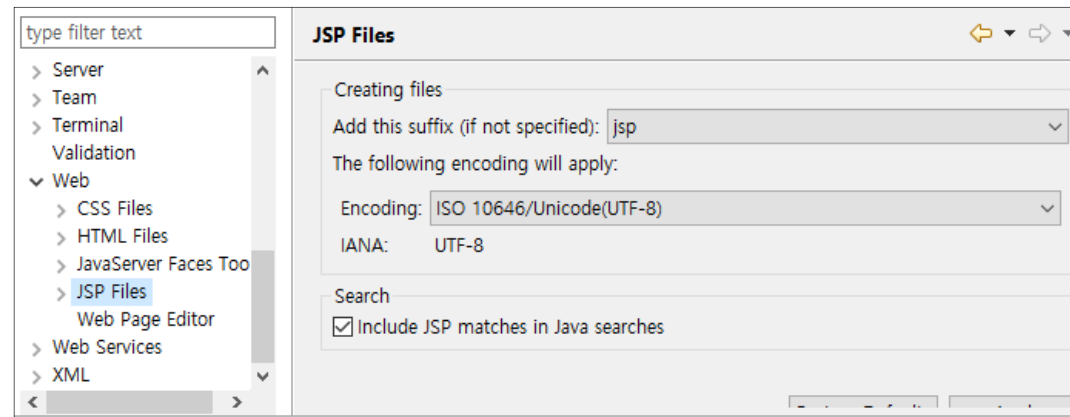
- 프로젝트 이름 : chapter04
 - ▶ web.xml 파일은 반드시 생성해야 함



- servlet-api 라이브러리 추가



- JSP 문서 문자 집합 지정
 - ▶ UTF-8로 지정



인클루드 지시어

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

menu.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3
4 [최신 뉴스]<HR>
5 12:32 - 자바 웹 프로그래밍 전면 개정판 출간 !!! <BR>
6 12:29 - 스프링프레임워크 적용 확산.<BR>
```

news.jsp

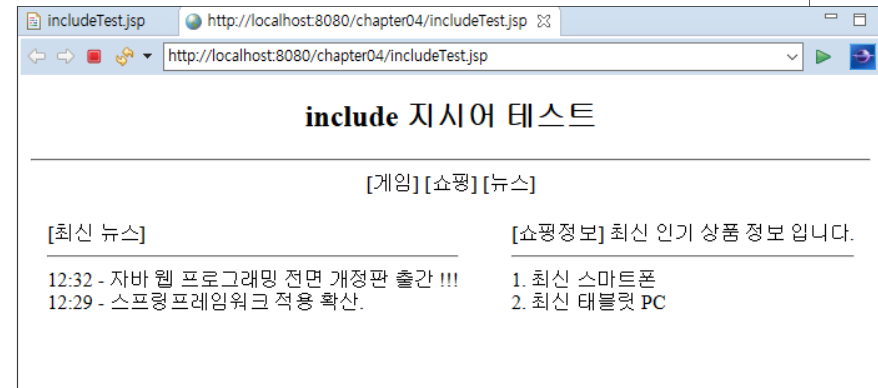
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3
4 [쇼핑정보] 최신 인기 상품 정보 입니다.<HR>
5 1. 최신 스마트폰 <BR>
6 2. 최신 태블릿 PC <BR>
```

shopping.jsp

인클루드 지시어

includeTest.jsp

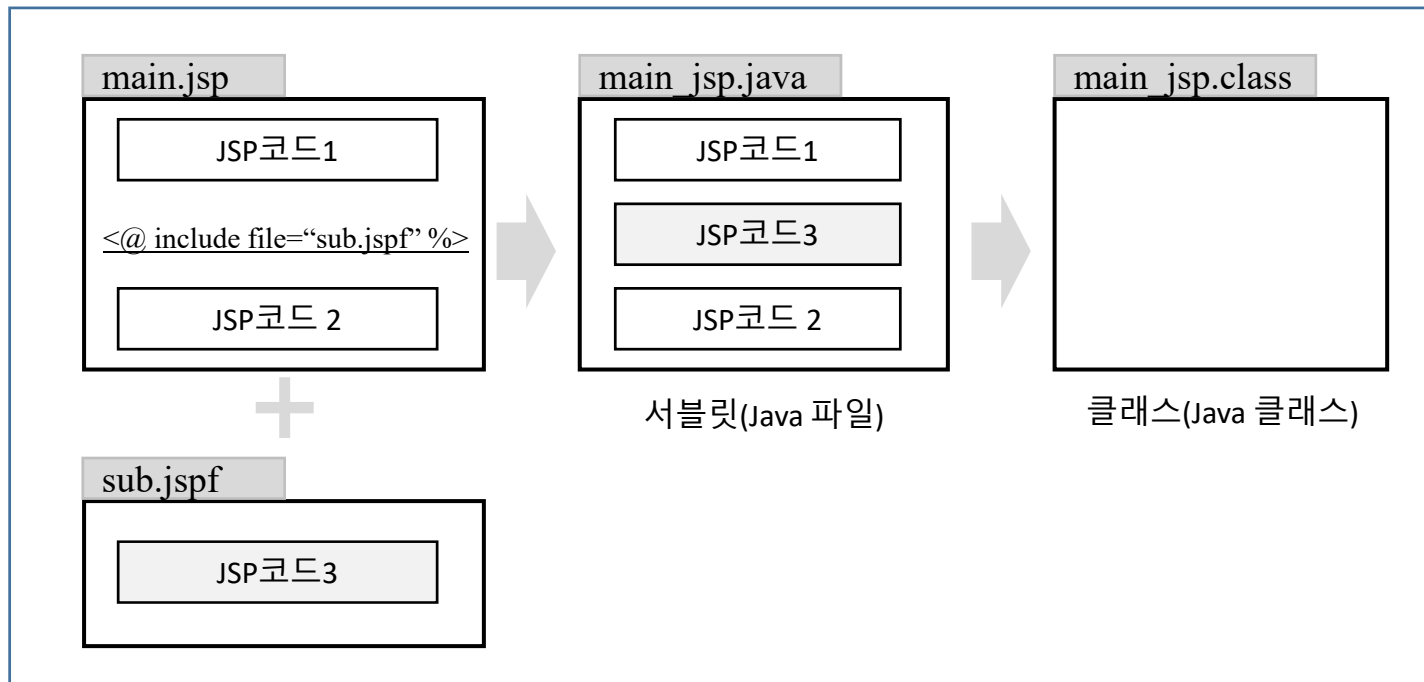
```
includeTest.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7 </head>
8 <body>
9   <div align="center">
10    <h2> include 지시어 테스트</h2>
11    <HR>
12
13    <%@ include file="menu.jsp" %>
14    <p>
15    <table border=0 cellpadding=5 cellspacing=1>
16      <tr>
17        <td><%@include file="news.jsp" %> </td>
18        <td width="20">&nbsp;</td>
19        <td><%@include file="shopping.jsp" %></td>
20      </tr>
21    </table>
22
23    </div>
24 </body>
25 </html>
```



인클루드 지시어

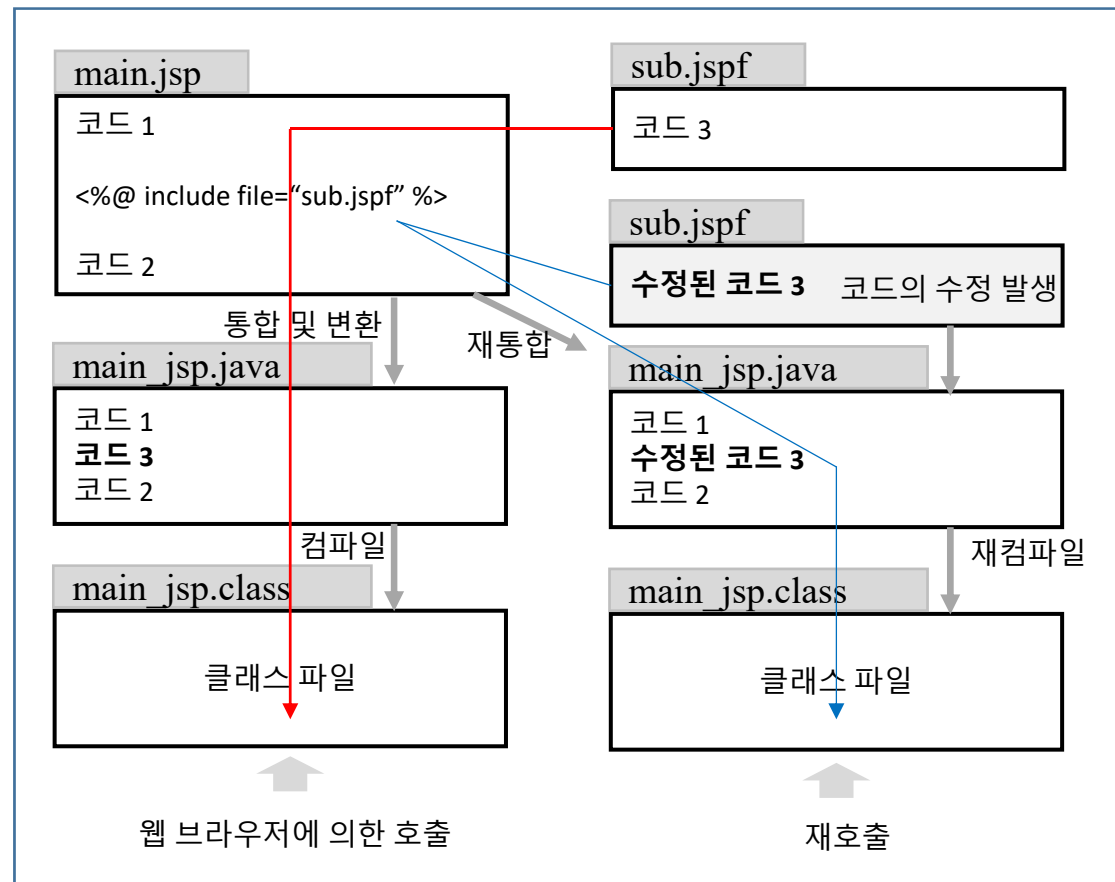
■ include 지시문의 처리 방식

- include되는 문서의 소스 코드 자체가 포함됨
 - ▶ 하나의 문서로 통합하여도 서블릿과 클래스 파일이 생성됨



인클루드 지시어

- 포함되는 문서의 소스코드가 변경되는 경우
 - 수정된 코드를 다시 포함시켜 다시 컴파일 됨



선언과 표현식

선언(declaration)

■ 선언

- JSP 내에서 멤버 변수나 메서드를 선언하는 것을 의미

```
<%!  
    멤버변수 선언;  
    public 리턴타입 메소드 이름(파라미터 리스트) {  
        Java 코드 리스트  
    }  
%>
```

```
<%!  
    String str = "test";  
    public boolean check() {  
        return false;  
    }  
%>
```

- ▶ 선언은 반드시 '<%!'로 시작해서 '%>'로 끝남

표현식

■ 표현식

- JSP에서는 브라우저에 데이터를 출력할 때 out 객체의 print(), println() 메서드를 사용
- 표현식은 간단한 데이터를 out 객체의 메서드를 사용하지 않고 출력하기 위해 사용

```
<%  
    String name="홍길동";  
    out.println( "당신의 이름은" + name + "입니다" );  
%>
```

```
<%  
    String name="홍길동";  
%>  
  
당신의 이름은 <%= name %>입니다;
```

- ▶ [주의] 표현식은 스크립트릿 내부에는 사용할 수 없음
- ▶ [주의] 표현식의 끝에는 세미콜론(;)을 사용하지 않음

표현식

■ 표현식의 사용

- 메서드 호출

- ▶ [예] `<%= calculator() %>`

- 변수의 내용을 출력하거나 연산(숫자나 문자열) 수행

- ▶ [예] `<%= result %>`

- ▶ [예] `<%= "i+2="+ (i+2) + "입니다" %>`

- `<FORM>` 태그를 사용해 다른 JSP 문서에 JSP 변수를 전달

- ▶ [예] `<FORM NAME="myform" METHOD=POST ACTION="list.jsp?data=<%=num%>">`

스크립트릿 (scriptlet)

스크립트릿의 개요

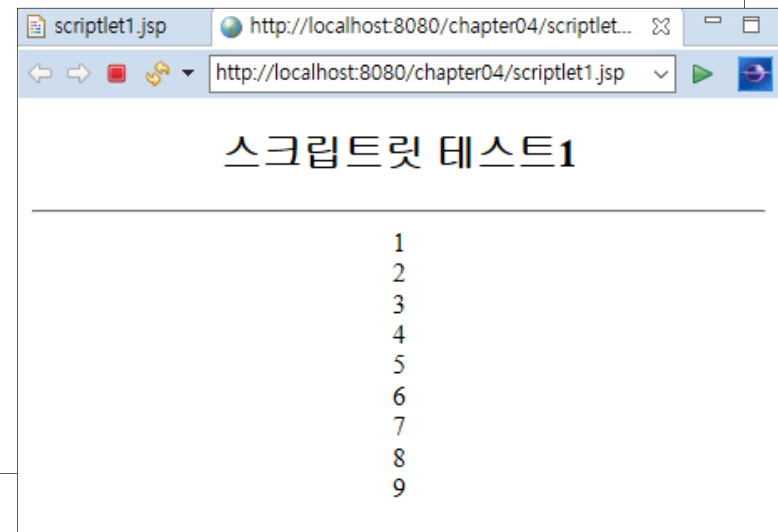
■ 스크립트릿(Scriptlet)

- JSP 문서 내에서 자바 코드를 기술할 수 있는 부분
 - JSP의 가장 큰 특징 중 하나로 대부분의 자바 클래스 라이브러리를 사용할 수 있음
- 스크립트릿 내부에는 자바 코드만 사용할 수 있음
 - HTML 코드를 사용하려면 `out` 객체의 `println()` 메서드 내에 표현해야 함
- 초기 JSP 발전에 큰 기여를 하였으나 지금은 JSP내에서 스크립트릿 사용은 권장되지 않음
 - 하나의 JSP 문서 내에서 화면 출력 내용과 Java코드가 섞여 있을 경우 유지보수가 어려움
 - 디자이너들과의 협업이 어려움
 - 표현 언어, JSTL, custom tag library, Beans 가 주로 사용됨
- MVC 패턴을 사용하면 스크립트릿은 사용은 많지 않음
 - JSP는 View의 역할을 수행
 - 서블릿, 클래스, 빈즈가 Controller와 Model 역할을 수행

스크립트릿의 예

scriptlet1.jsp

```
scriptlet1.jsp
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 </head>
8 <body>
9
10   <div align="center">
11     <h2>스크립트릿 테스트1</h2>
12     <hr>
13
14   <%
15       for(int i=1;i<10;i++) {
16         out.println(i+"<br>");
17       }
18   %>
19
20   </div>
21
22 </body>
23 </html>
```

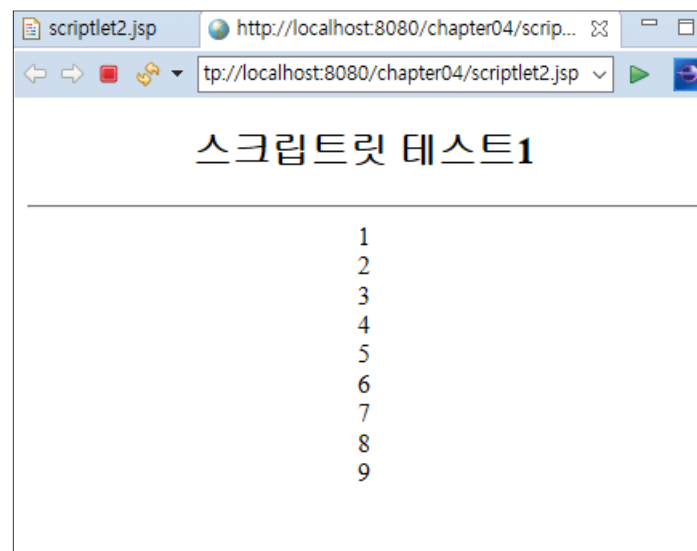


스크립트릿의 예

scriptlet2.jsp

scriptlet2.jsp

```
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 </head>
8 <body>
9
10   <div align="center">
11     <h2>스크립트릿 테스트1</h2>
12     <hr>
13
14   <%
15     for(int i=1;i<10;i++) {
16
17     %>
18     <%= i %> <br>
19
20   <%
21     }
22   %>
23
24   </div>
25
26 </body>
27 </html>
```

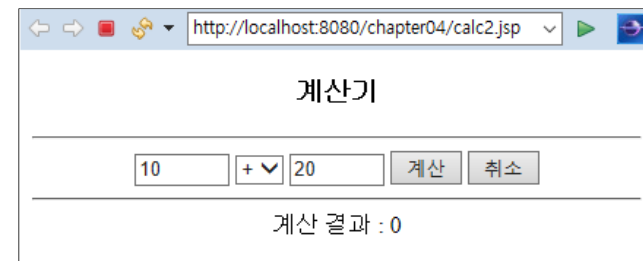
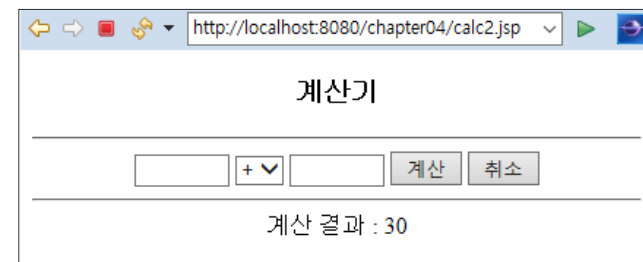


```
calc2.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%!
4   int num1 = 0;
5   int num2 = 0;
6   int result = 0;
7   String op = "";
8
9   public int calculate() {
10     if(op.equals("+")) {
11       result = num1 + num2;
12     } else if(op.equals("-")) {
13       result = num1 - num2;
14     } else if(op.equals("*")) {
15       result = num1 * num2;
16     } else if(op.equals("/")) {
17       result = num1 / num2;
18     }
19     return result;
20   }
21 %>
22
23 <%
24   if(request.getMethod().equals("POST")) {
25     num1 = Integer.parseInt(request.getParameter("num1"));
26     num2 = Integer.parseInt(request.getParameter("num2"));
27     op = request.getParameter("operator");
28   }
29 %>
```

```

31 <!DOCTYPE html>
32 <html>
33 <head>
34     <meta charset="UTF-8">
35 </head>
36 <body>
37
38     <div align=center>
39     <h3>계산기</h3>
40     <hr>
41
42     <form name="form1" method=post action="calc2.jsp">
43         <input type=text name="num1" size=5>
44         <select name="operator">
45             <option selected>+</option>
46             <option>-</option>
47             <option>*</option>
48             <option>/</option>
49         </select>
50         <input type=text name="num2" size=5>
51         <input type=submit value="계산">
52         <input type=reset value="취소">
53     </form>
54
55     <hr>
56     계산 결과 : <%=calculate()%>
57 </div>
58
59 </body>
60 </html>

```

주석문

■ 주석문

• JSP 주석문 사용 가능

```
<%-- JSP 주석문의 내용 --%>
```

- ▶ 스크립트릿 내에는 사용할 수 없음
- ▶ JSP 주석문 내에 또 다른 JSP 주석문을 중첩하여 사용할 수는 없음

• JAVA 주석문 사용 가능

```
// 단일 라인 주석문  
/* 다중 라인 주석문 */  
/** 선언문 내의 주석문 */
```

- ▶ 스크립트릿, 선언문 내에서 사용 가능

• HTML 주석문 사용 가능

```
<!-- HTML 주석문 -->
```

수고하셨습니다