

JSP 기본 문법

524730
2022년 봄학기
3/16/2022
박경신

JSP Syntax

- 주석(Comment) `<%-- 주석 --%>`
- 스크립트릿(Scriptlet) `<% 스크립트 %>`
- 선언(Declaration) `<%! 선언 %>`
- 표현식(Expression) `<%= 표현식 %>`
- 지시어(Directive) `<%@ 지시어 속성="값" %>`
 - page, include, taglib
- 액션 태그(Actions) `<jsp:include ...>`
 - include, forward, param, useBean, setProperty, getProperty, plugin, element, attribute, body, text

JSP Syntax

- 내장 객체(Implicit Objects)
 - request, response, out, session, application, config, pageContext, page, exception
- JSP Java Beans
- JSP Database Access
- JSP Form Processing
- JSP Filters, Cookies, Session
- JSP Exception Handling
- JSP Expression Language
- JSTL (JSP Standard Tag Library)
 - Core tags, Formatting tags, SQL tags, XML tags, JSTL Functions
- JSP Custom tags
 - Custom tags with body in JSP, Custom tags with attributes

Comment

□ 자바 주석

- JSP 스크립트릿 영역 내에서 사용함
- 기본형식: `/* */`, `//`, `/** */`

□ HTML 형식의 주석

- JSP는 HTML과 함께 구성되므로 HTML 형식의 주석 사용 가능
- 단, JSP 문법이 사용된 부분에는 사용 불가
- 기본형식: `<!-- 주석 -->`

□ JSP 주석

- HTML과 JSP 문법 모든 부분에 사용 가능
- 소스 보기를 하면 보이지 않음
- 기본형식: `<%-- 주석 --%>`

Scriptlet

□ 스크립트릿

- JSP 파일 중간에 자바 코드를 기술함.
- 태그라이브리 등을 권장하며 스크립트 남용은 페이지 가독성을 떨어뜨리고 유지보수를 어렵게 함.
- 스크립트릿에서 화면 출력을 위해서 `out.println()`을 사용.
- 스크립트릿을 닫고 HTML 출력 후 다시 스크립트릿 작성도 가능.
- 기본형식: **<% 자바 코드 %>**

```
<%  
    // 로컬변수 선언이나 프로그램 로직이 올 수 있다.  
    String str = "test";  
    for(int i=0;i <10;i++) {  
        out.println(i);  
    }  
    %>
```

Declaration

□ 선언

- JSP 파일에서 메서드나 멤버 변수를 쓸 때 사용
- JSP 내에 프로그램 코드를 많이 두는 것은 바람직하지 않음.
- 기본형식: **<%! 멤버 변수 선언 또는 메소드 선언 %>**

```
<%!  
  // 멤버변수 선언이나 메서드 선언이 올수 있다.  
  String str = "test";  
  public boolean check() {  
    return false;  
  }  
%>
```

Expression

□ 표현식

- JSP 파일에서 간단한 출력을 위해 사용함.
- `out.println()` 의 인자로 적합한 형태여야 함.
- 기본형식: **<%= 메소드 호출 및 변수 %>**

메서드 호출 : `<%= calculator() %>`

변수 출력 : `<%= result %>`

사칙 연산과 문자열 결합 : `<%= "i+2=" + i + 2 + " 입니다" %>`

Directives

□ 지시어 (Directives)

- page 지시어 `<%@ page 속성1="속성값1" 속성2="속성값2"..%>`
- include 지시어 `<%@ include file="포함할 파일" %>`
- taglib 지시어 `<%@ taglib uri="태그라이브러리 위치" prefix="이름" %>`

page 지시어

□ page 지시어

- 현재 JSP 페이지를 컨테이너에서 처리하기 위한 각종 속성 지정
- 기본 형식

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

- 사용 예

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

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

page 지시어 속성

속성	설명	기본값
language	스크립트 언어를 지정	Java
import	JSP 파일 내에서 사용할 외부 자바 패키지나 클래스 지정	
session	세션 생성 여부 지정	True
buffer	버퍼 크기 지정	8kb
autoFlush	버퍼 내용 자동 비움 지정	True
isThreadSafe	단일 스레드 모델을 사용하여 동시성 제어 여부 지정	true
info	JSP 페이지 설명	
errorPage	에러 발생시 호출 페이지 지정	
isErrorPage	에러만 처리하는 페이지 지정	false
contentType	MIME 형식과 캐릭터셋 설정	text/html; charset =ISO-8859-1

include 지시어

□ include 지시어

- 현재 JSP 파일에 다른 JSP나 HTML 문서를 포함시킴
- 모든 JSP 페이지에서 사용되는 변수 지정 또는 모든 JSP 페이지에서 중복되는 문장 등에 활용
- 기본 형식

<%@ include file="포함할 파일" %>

```
<%@ page contentType="text/html;charset=euc-kr" %>
<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
<H2> include 지시어 테스트 1 </H2>
<HR>
<%@ include file="footer.jsp" %>
</BODY>
</HTML>
```

taglib 지시어

□ taglib 지시어

- 현재 JSP 페이지에 사용할 커스텀 태그 라이브러리를 지정함.
- 표준화된 커스텀태그 : **JSTL(JSP Standard Tag Library)**
- JSP 페이지 내에 불필요한 자바 코드를 줄일 수 있음.
- 기본 형식

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

```
<%@ page contentType="text/html;charset=euc-kr" %>
<%@ taglib uri="/META-INF/mytag.tld" prefix="mytag" %>
<HTML>
<HEAD><TITLE> </TITLE> </HEAD>
<BODY>
<mytag:GetInfo name="dinfree" />
</BODY>
</HTML>
```

액션 태그

□ 액션 태그

- 서버나 클라이언트에게 어떤 행동을 하도록 명령하는 태그
- JSP 페이지에서 페이지와 페이지 사이를 제어하거나
`<jsp:forward .. />`,
- 다른 페이지의 실행 결과 내용을 현재 페이지에 포함하거나
`<jsp:include .. />`,
- 자바 빈즈(JavaBeans) 등의 다양한 기능을 제공 `<jsp:useBean .. />`

- XML 형식 `<jsp: ... />`를 사용

액션 태그 종류

액션	사용법	기능
include	<pre><jsp:include page="xx.jsp" /></pre>	다른 페이지를 현재 페이지에 포함
forward	<pre><jsp:forward page="xx.jsp" /></pre>	현재 페이지의 제어를 다른 페이지로 전달
param	<pre><jsp:param /></pre>	<pre><jsp:include></pre> , <pre><jsp:forward></pre> , <pre><jsp:plugin></pre> 에 인자를 추가
useBean	<pre><jsp:useBean scope="page" id="cls" class="xx.MyBean" /></pre>	xx 패키지의 MyBean 클래스를 cls로 page 범위에서 사용할 것을 선언
setProperty	<pre><jsp:setProperty name="cls" property="xxx" /></pre>	useBean으로 선언된 빈즈 클래스의 setXxx() 메서드를 호출
getProperty	<pre><jsp:getProperty name="cls" property="xxx" /></pre>	useBean으로 선언된 빈즈 클래스의 getXxx() 메서드를 호출

액션 태그 종류

액션	사용법	기능
plugin	<code><jsp:plugin /></code>	웹 브라우저에 자바 애플릿을 실행. 자바 플러그인에 대한 OBJECT 또는 EMBED 태그를 만드는 브라우저별 코드를 생성
element	<code><jsp:element /></code>	동적 XML 요소를 설정
attribute	<code><jsp:attribute /></code>	동적으로 정의된 XML 요소의 속성을 설정
body	<code><jsp:body /></code>	동적으로 정의된 XML 요소의 몸체를 설정
text	<code><jsp:text /></code>	JSP 페이지 및 문서에서 템플릿 텍스트를 작성

include 액션 태그

□ include 액션 태그

- include 지시어와 유사하게 현재 JSP 페이지의 특정 영역에 다른 페이지를 import함.
- include 지시어와의 차이점
 - 실행 시점에 해당 파일을 호출하고 결과를 포함시킴.
 - 파라미터를 전달할 수 있음.
- 기본 형식

<jsp:include page="포함할 파일명" />

→ include_action.jsp

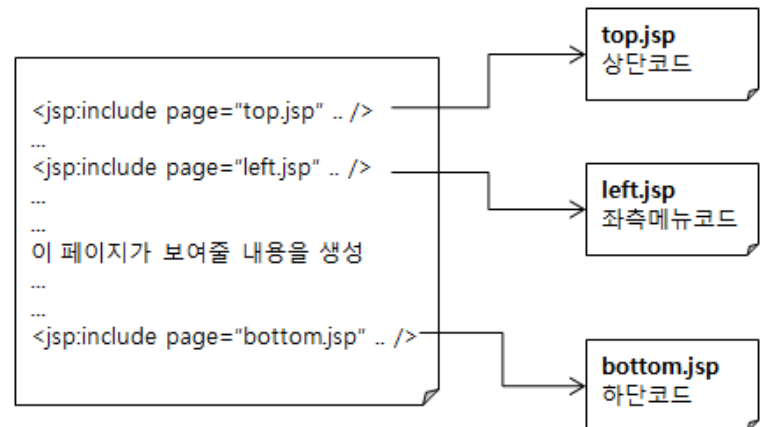
```
<jsp:include page="footer2.jsp">  
  <jsp:param name="email" value="test@test.net" />  
  <jsp:param name="tel" value="000-000-0000" />  
</jsp:include>
```

→ **footer2.jsp**

```
<%= request.getParameter("email") %>,
```


include 액션 태그

- page 속성 값
 - 현재 JSP 페이지 내에 포함할 내용을 가진 **외부 파일명**
 - 포함할 수 있는 외부 파일은 HTML, JSP, 서블릿 등.
 - 외부 파일은 현재 JSP 페이지와 같은 디렉터리에 있으면 파일명만 설정하고, 그렇지 않으면 전체 URL(또는 상대 경로)을 설정해야 함
- flush 속성 값
 - 설정한 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장한 결과를 처리, 기본 값은 false
 - true 로 설정하면 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장된 내용을 웹 브라우저에 출력하고 출력 버퍼를 비움
- 중복 영역을 모듈화하는데 유용



include 액션태그 와 include 지시어

비교항목	<jsp:include>	include 지시어
처리시간	요청 시간에 처리	JSP 파일을 자바 소스로 변환할 때 처리
기능	별도의 파일로 요청 처리 흐름을 이동	현재 파일에 삽입시킴
데이터 전달방법	request 기본 객체나 <jsp:param>을 이용한 파라미터 전달	페이지 내의 변수를 선언한 후, 변수에 값 저장
용도	화면의 레이아웃의 일부분을 모듈화할 때 주로 사용된다.	다수의 JSP 페이지에서 공통으로 사용되는 변수를 지정하는 코드나 저작권과 같은 문장을 포함한다.

forward 액션 태그

□ forward 액션 태그

- include 액션과 유사하지만 현재 페이지의 **제어권을 완전히 다른 페이지로 전달함**
- 브라우저 URL에는 최종 전달된 파일명이 보이는 것이 아니라 최초 호출한 파일명이 보임. 즉, 실행 시점에 해당 파일을 호출하고 결과를 포함시킴.
- 기본 형식

```
<jsp:forward page="포워딩할 파일" />
```

→ include_action.jsp

```
<jsp:forward page="footer2.jsp">
```

```
  <jsp:param name="email" value="test@test.net" />
```

```
  <jsp:param name="tel" value="000-000-0000" />
```

```
</jsp:forward>
```

→ footer2.jsp

```
<%= request.getParameter("email") %>,
```

forward 액션 태그

□ forward 액션 태그의 전형적인 사용법

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%
    String forwardPage = null;

    // 조건에 따라 이동할 페이지를 지정
    if (조건판단1) {
        forwardPage = "페이지|URI1";
    } else if (조건판단2) {
        forwardPage = "페이지|URI2";
    } else if (조건판단3) {
        forwardPage = "페이지|URI3";
    }
%>
<jsp:forward page="<%= forwardPage %>" />
```

param 액션 태그

□ param 액션 태그

- 현재 JSP 페이지에서 다른 페이지에 정보를 전달하는 태그
- 이 태그는 단독으로 사용되지 못하며 **<jsp:include>**, **<jsp:forward>**, **<jsp:plugin>** 태그의 내부에서 사용
- 다른 페이지에 여러 개의 정보를 전송해야 할 때는 다중의 param 액션 태그를 사용
- 기본 형식

```
<jsp:include page="/module/top.jsp" flush="false" >  
  <jsp:param name="param1" value="value1" />  
  <jsp:param name="param2" value="value2" />  
</jsp:include>
```

useBean 액션 태그

□ useBean 액션 태그

- 빈즈 컴포넌트와의 상호작용 지원
- JSP 페이지에서 자바빈즈를 사용하기 위해 실제 자바 클래스를 선언하고 초기화하는 태그
- id 속성과 scope 속성을 바탕으로 자바빈즈의 객체를 검색하고, 객체가 발견되지 않으면 빈 객체를 생성
- 기본 형식

```
<jsp:useBean id="이름" class="패키지.클래스" scope="범위" />
```

```
<jsp:useBean id="person" class="com.dto.Person" scope="request" />
```

```
아이디: <%= person.getId() %>
```

```
이름: <%= person.getName() %>
```

setProperty 액션 태그

□ setProperty 액션 태그

- setProperty 액션태그는 프로퍼티의 값 지정하기
- useBean 액션 태그와 함께 자바빈즈의 **setter()** 메소드에 접근하여 자바빈즈의 멤버 변수인 프로퍼티의 값을 저장하는 태그
- 폼 페이지로부터 전달되는 요청 파라미터의 값을 직접 저장하거나 자바빈즈의 프로퍼티로 변경하여 값을 저장 가능
- 모든 자바빈즈 프로퍼티 이름과 동일하게 요청 파라미터를 설정가능
- 기본 형식

```
<jsp:setProperty name="이름" property="속성" value="값" />
```

```
<jsp:useBean id="person" ... />  
<jsp:setProperty name="person" property="name" value="Park" />  
<%  
    out.println("이름: " + person.getName());  
%>
```

setProperty 액션 태그

- setProperty 액션 태그
 - 요청 parameter 이름과 자바빈즈의 프로퍼티 이름이 일치하는 경우

→ personInputForm.jsp

```
<form action="personProcess.jsp" method="post">  
  <input name="name" value="Park" />  
</form>
```

→ personProcess.jsp

```
<jsp:setProperty name="person" property="name" />
```


setProperty 액션 태그

□ setProperty 액션 태그

- 요청 parameter 이름과 자바빈즈의 프로퍼티 이름이 일치하지 않는 경우

→ personInputForm.jsp

```
<form action="personProcess.jsp" method="post">  
  <input name="userName" value="Park" />  
</form>
```

→ personProcess.jsp

```
<jsp:setProperty name="person" property="name" param="userName" />
```

setProperty 액션 태그

- setProperty 액션 태그
 - 요청 parameter 이름과 자바빈즈의 프로퍼티 이름이 모두 일치하는 경우

→ personInputForm.jsp

```
<form action="personProcess.jsp" method="post">  
  <input name="id" value="10" />  
  <input name="name" value="Park" />  
</form>
```

→ personProcess.jsp

```
<jsp:setProperty name="person" property="*" />
```

getProperty 액션 태그

□ getProperty 액션 태그

- getProperty 액션태그는 프로퍼티의 값 가져오기
- 기본 형식

```
<jsp:getProperty name="이름" property="속성" />
```

```
<jsp:useBean id="person" class="com.dto.Person" scope="request" />  
아이디: <jsp:getProperty name="person" property="id" />  
이름: <jsp:getProperty name="person" property="name" />
```

plugin 액션 태그

□ plugin 액션 태그

- 웹 브라우저에서 자바 플러그인을 사용할 수 있게 함
- 애플릿이나 자바 빈즈 컴포넌트 실행 지원.
- 기본 형식

```
<jsp:plugin type="애플릿/자바빈" code="클래스"  
codebase="디렉토리" >
```

```
</jsp:plugin>
```

element, attribute, body 액션 태그

□ element 액션 태그

- 동적으로 XML element를 지정
- 기본 형식

```
<jsp:element name="요소이름"> </jsp:element>
```

□ attribute 액션 태그

- 동적으로 XML element의 attribute를 지정
- 기본 형식

```
<jsp:attribute name="속성이름"> </jsp:attribute>
```

□ body액션 태그

- 동적으로 XML element의 body를 지정
- 기본 형식

```
<jsp:body>내용</jsp:body>
```

```
<jsp:element name = "xmlElement">  
  <jsp:attribute name = "xmlElementAttr">  
    attribut의 값  
  </jsp:attribute>  
  <jsp:body>  
    XML element의 body  
  </jsp:body>  
</jsp:element>
```

text 액션 태그

- text 액션 태그

- JSP 페이지의 템플릿 텍스트 지정
- 기본 형식

`<jsp:text>` 템플릿 데이터 `</jsp:text>`