

Action Tag & EL & JSTL





Action Tag



Action Tag

JSP Action Tag

- XML 기술을 이용하여 기존의 JSP 문법을 확장하는 메커니즘을 제공하는 태그
- 웹 브라우저에서 실행되는 것이 아니라 웹 컨테이너에서 실행되고 결과만 브라우저 출력

	표준 액션 태그	커스텀 액션 태그
사용법	<ul style="list-style-type: none">- JSP 페이지에서 바로 사용- 태그 앞에 jsp 접두어가 붙음	<ul style="list-style-type: none">- 별도의 라이브러리 설치 필요- 라이브러리 선언에 맞는 접두어가 붙음
예시	<code><jsp:include page="header.jsp"/></code>	<code><c:set var="cnt" value="0"/></code>

※ 접두어란 태그 이름앞에 ooo: 형식으로 제공하는 태그의 그룹을 지정하는 것



Action Tag

표준 액션 태그

- JSP에서 기본으로 제공하는 태그

태그 이름	설명
jsp:include	현재 페이지에 특정 페이지를 포함 할 때 사용
jsp:forward	현재 페이지 접근 시 특정 페이지로 이동 시킬 때 사용
jsp:param	<jsp:incpude>,<jsp:forward>의 하위 요소로 사용되며, 해당 페이지에 전달할 값을 기록할 때 사용
jsp:useBean	JavaBean 객체를 사용하기 위한 태그
jsp:setProperty	Java 객체 사용 시 Setter 역할과 동일
jsp:getProperty	Java 객체 사용 시 Getter 역할과 동일



Action Tag

jsp:include

- `<%@ include file="파일명" %>`과 쓰임이 동일하나, jsp 파일이 java 파일로 변환 될 때 삽입되는 `<%@ include %>`와 달리 `<jsp:include>`는 jsp파일이 java 파일로 바뀌고 컴파일이 완료되어 런타임 시 삽입 된다.

사용 방식

`<jsp:include page="파일명" />`

`<jsp:include page="./header.jsp" />`



Action Tag

jsp:forward

- 하나의 JSP 페이지에서 다른 JSP 페이지로 요청 처리를 전달할 때 사용
- 전달하는 페이지에서 request, response 객체가 같이 전달되며 URL은 변경되지 않음

사용 방식

```
<jsp:forward page="파일명" />  
<% if(str.equals("A")) { %>  
    <jsp:forward page="A_Page.jsp" />  
<% } else { %>  
    <jsp:forward page="B_Page.jsp" />  
<% } %>
```



Action Tag

jsp:useBean

- java class를 참조하여 빈 객체를 생성하고, setProperty와 getProperty를 통해 값을 저장, 조회할 수 있으며, 이미 같은 이름의 객체가 생성된 경우 기존 객체를 참조

사용 방식

```
<jsp:useBean id="객체명" class="패키지명.클래스명 " scope="범위"/>
<jsp:useBean id="m" class="model.vo.Member " scope="request">
    <jsp:setProperty name="m" property="memberName"
value="홍길동" />
</jsp:useBean>
```





EL



EL

EL(Expression Language)

- JSP 2.0 버전에서 추가된 것으로 `<%= %>`, `out.print()`와 같이 JSP에 쓰이는 JAVA코드를 간결하게 사용하는 방법으로, 화면에 표현하고자 하는 코드를 `${value}`의 형식으로 표현하여 작성하는 것

사용 방식

```
<%= request.getParameter("name") %>
```

```
${param.name}
```



EL 연산자 기호

종류	일반 연산자	EL 기호 연산자
덧셈, 뺄셈	+, -	+, -
곱셈, 나눗셈	*, /	*, div
나머지 연산	%	mod
and, or 연산	&&,	and, or
! 연산	!	not
~ 보다 작다	<	lt(less than)
~ 보다 크다	>	gt(greater than)
작거나 같다	<=	le(less or equal)
크거나 같다	>=	ge(greater or equal)
~와 같다	==	eq(equal)
~와 다르다	!=	ne(not equal)
null 값 처리	value == null	empty



EL

EL 연산자 우선 순위

순위	표기법
1	<code>[]</code> , <code>.</code>
2	<code>()</code>
3	<code>not</code> , <code>!</code> , <code>empty</code>
4	<code>&</code> , <code>/</code> , <code>div</code> , <code>%</code> , <code>mod</code>
5	<code>+</code> , <code>-</code>
6	<code><</code> , <code><=</code> , <code>></code> , <code>>=</code> , <code>lt</code> , <code>le</code> , <code>gt</code> , <code>ge</code>
7	<code>==</code> , <code>!=</code> , <code>eq</code> , <code>ne</code>
8	<code>&&</code> , <code>and</code>
9	<code> </code> , <code>or</code>
10	<code>?:</code> (삼항 연산자)



EL 내장 객체

객체 명	설명
pageScope	page 영역의 객체에 접근
requestScope	request 영역의 객체에 접근
sessionScope	session 영역의 객체에 접근
applicationScope	application 영역의 객체에 접근
param	전달된 파라미터 값을 받아올 때 사용
paramValues	전달된 파라미터들을 배열로 받아올 때 사용
header	사용자의 특정 헤더 정보를 받아올 때 사용
headerValues	사용자의 헤더 정보를 배열로 받아올 때 사용
cookie	\${cookie.key 명}으로 쿠키값을 조회
iniParam	초기 파라미터를 조회
pageContext	pageContext 경로를 조회





JSTL



JSTL

JSTL(JSP Standard Tag Library)

- JSP에서 사용하는 커스텀 태그로, 공통으로 사용하는 코드의 집합을 사용하기 쉽게 태그화하여 표준으로 제공한 것

사용 방식

1. 라이브러리 등록
2. 선언 `<%@ taglib uri=http://java.sun.com/jsp/jstl/core" prefix="c" %>`
3. 사용 `<c:out value="${welcome}"/>`



JSTL

JSTL 태그 종류

종류	표기법
Core Tags	<ul style="list-style-type: none">- 변수와 url, 조건문, 반복문 등의 로직과 관련된 JSTL 문법 제공- <%@ taglib url="http://java.sun.com/jsp./jstl/core" prefix="c" %>
Formatting Tags	<ul style="list-style-type: none">- 메시지 형식이나 숫자, 날짜 형식과 관련된 포맷 방식을 제공- <%@ taglib url="http://java.sun.com/jsp./jstl/fmt" prefix="fmt" %>
Function Tags	<ul style="list-style-type: none">- trim, substring과 같은 문자열 처리 함수를 제공- <%@ taglib url="http://java.sun.com/jsp./jstl/function" prefix="fn" %>
XML Tags	<ul style="list-style-type: none">- 데이터의 XML 파싱 처리 등 XML 문서를 다루는 함수를 제공- <%@ taglib url="http://java.sun.com/jsp./jstl/xml" prefix="x" %>
SQL Tags	<ul style="list-style-type: none">- 페이지 내에서 DB를 연동하고, 쿼리를 실행할 수 있는 함수 제공- <%@ taglib url="http://java.sun.com/jsp./jstl/sql" prefix="sql" %>

※ SQL Tags는 DB를 직접 연동하는 것에 용이하나, DB 트랜잭션을 별도로 묶어서 관리하는 MVC Model 2 방식에는 맞지 않기 때문에 사용을 지양한다.





core tags



JSTL Core Tags

<c:set>

- 변수를 선언하고 그 변수에 초기값을 대입하는 기능의 태그
- 자바 변수 선언 방식과 유사
- scope 속성을 이용하여 속성 저장 가능(설정하지 않는 경우 page임)
- 자바 변수 선언 후 초기값 대입
`int num = 100`
- <c:set> 변수 선언 후 초기값 대입
`<c:set var="num" value="100" scope="request" />`



JSTL Core Tags

<c:set> 사용법

- 변수 타입을 별도로 선언하지 않음
- 초기값을 반드시 입력
- <c:set>으로 선언한 변수는 EL식 안에서 사용가능
- <c:set>으로 선언한 변수는 스크립틀릿 요소에서는 사용 불가
- 스크립틀릿에서 선언된 변수는 <c:set>에서 사용 가능

```
<% int num1=10, num2=20; %>
```

```
<c:set var="sum" value="<%= num1+num2 %>" />
```

```
${sum}
```



JSTL Core Tags

`<c:remove>`

- `<c:set>`을 이용해서 선언한 변수는 page, request, session, application영역에 속성으로 저장되기 때문에 이를 삭제하기 위한 태그

```
<c:remove var="num" scope="request" />
```



JSTL Core Tags

<c:out>

- <c:out>태그는 데이터를 출력할 때 사용하는 태그
- <, >, & 특수 문자를 자동으로 이스케이프 시퀀스(escape sequence)로 처리
※ default : true

**<c:out value="글씨를 진하게 하려면짼 태그를 사용하면 됨"
escapeXml="false" />**

**<c:out value="글씨를 진하게 하려면짼 태그를 사용하면 됨"
escapeXml="true" />**



JSTL Core Tags

<c:if>

- 자바의 if문과 비슷한 역할
- <c:if>태그에서 조건식은 test라는 속성의 값으로 지정하며 반드시 EL형식

```
<c:if test="${num1 > num2}">  
    num1이 더 큼니다!
```

```
</c:if>
```



JSTL Core Tags

<c:choose>

- 자바의 switch(if ~ else if)문과 비슷한 역할을 하는 태그
- <c:when>,<c:otherwise> 태그와 함께 사용되는데, 각각 switch 문의 case와 default절과 비슷한 역할을 수행

<c:choose>

```
<c:when test="{num == 0}">
```

num은 0 입니다.

```
</c:when>
```

```
<c:when test="{num == 1}">
```

num은 1 입니다.

```
</c:when>
```

```
<c:otherwise>
```

num은 0도 1도 아닙니다.

```
</c:otherwise>
```

```
</c:choose>
```



JSTL Core Tags

<c:forEach>

- 자바의 for, for-eachans에 해당하는 기능을 제공

속성	설명
items	- 반복할 객체 명(Collection 객체)
begin	- 반복이 시작할 요소 번호(0 ~ n)
end	- 반복이 끝나는 요소 번호
step	- 반복할 횟수 번호
var	- 현재 반복 횟수에 해당하는 변수의 이름
varStatus	- 현재 반복에 해당하는 객체의 요소



JSTL Core Tags

<c:forEach>

varStatus 속성

속성	설명
current	- 현재 반복 횟수의 객체
index	- 반복 라운드의 제로기반 인덱스(0 ~ n-1)
count	- 반복 라운드의 1기반 인덱스(1 ~ n)
first	- 현재 라운드가 반복을 통한 첫번째임을 의미
last	- 현재 라운드가 반복을 통한 마지막 번째임을 의미



JSTL Core Tags

<c:forEach> 사용 예

```
<c:forEach items="${bookList}" var="book" varStatus="status">  
  <tr>  
    <td><c:out value="${status.count}" /></td>  
    <td><c:out value="${book.name}" /></td>  
  </tr>  
</c:forEach>
```



JSTL Core Tags

`<c:forTokens>`

- 문자열에 포함된 구분자를 통해 토큰을 분리해서 반복 처리
- `items` 속성에는 토큰을 포함하는 문자열, `delims` 속성에는 토큰 구분자를 기술

```
<c:forTokens items="yellow blue pink" var="color" delims=" "
    ${color} <br>
</c:forTokens>
```



JSTL Core Tags

`<c:url>`

- url 경로를 생성하고, 해당 url의 param속성을 선언하여 쿼리스트링을 정의할 수 있는 태그
- 쿼리스트링을 미리 정의하여 제어

```
<c:url var="url" value="/jstl1.jsp">  
    <c:param name="name" value="abc" />  
</c:url>
```

```
<a href="{url}">JSTL페이지로 이동</a>
```

※ `JSTL페이지로 이동`과 동일

