JSP 란?

JSP 는 JavaServerPage 의 약어로 동적 페이지를 작성하는데 사용되는 자바의 표준 기술입니다. HTML 페이지가 정적인 기능을 담당했다면, JSP 페이지는 HTML 의 정적인 기능을 포함하여 동적인 기능까지 담당합니다.

디렉티브(Directive)

디렉티브는 JSP 페이지에 대한 설정 정보를 지정할 때 사용되며, 다음과 같은 구문을 통해서 디렉티브를 선언할 수 있습니다.

```
<%@ 디렉티브이름 속성 1="값 1" 속성 2="값 2" ... %>
```

그러면 실제 JSP 에서 사용되는 디렉티브의 형태를 보도록 합시다.

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

여기서 디렉티브의 이름은 "page"가 되고, "contentType"이라는 속성을 사용했으며 그 값으로는 "text/html; charset=utf-8"을 사용하고 있습니다.

위의 코드의 의미는 JSP 페이지의 내용이 html 형식으로 생성된다는 것이고, 응답결과를 보여줄 때 사용할 문자의 인코딩 타입이 utf-8 형식이라는 의미 입니다.

디렉티브	설명
page	JSP 페이지에 대한 정보를 지정한다.
taglib	JSP 페이지에서 사용할 태그 라이브러리를 지정한다.
include	JSP 페이지의 특정 영역에 다른 문서를 포함시킨다.

영역(scope)

확인: 영역이라는 한글보다는 scope 라는 영문을 주로 사용하니 앞으로는 scope 라고 명칭하겠습니다.

웹 어플리케이션은 다음의 네 가지 scope 로 구성됩니다.

- **Page scope** : 하나의 JSP 페이지를 처리할 때 사용되는 scope
- **Request scope** : 하나의 HTTP 요청을 처리할 때 사용되는 scope
- **Session scope** : 하나의 웹 브라우저오 관련된 scope
- **Application scope** : 하나의 웹 어플리케이션과 관련된 scope

Page scope 는 클라이언트의 요청에 의해 실행된 JSP 페이지입니다.

Request scope 는 한 번의 웹 브라우저 요청과 관되어 있습니다. 웹 브라우저가 요청을 보낼 때마다 새로운 Request scope 가 생성됩니다. Page scope 는 오직 하나의 JSP 페이지만을 포함하는데 반해, Request scope 는 하나의 요청을 처리하는데 사용되는 모든 JSP 페이지를 포함합니다.

Session scope 는 실행중인 브라우저가 종료되지 않는한 유지되는 영역입니다. 다만, 코드를 통해 강제적으로 제어할 수 있습니다.

Application scope 는 실행중인 웹 어플리케이션이 종료되지 않는한 유지되는 영역입니다. 이또한 Session scope 와 마찬가지로 코드를 통해 강제적으로 제어할 수 있습니다.

표현 언어(Expression Language-EL)

EL 을 사용하면 JSP 에서 간결한 코드를 사용해서 값을 출력할 수 있습니다.

EL 은 다음과 같이 $와 괄호'{}'그리고 표현식을 사용하여 값을 표현합니다.

${표현식}

위의 표현식 부분에는 EL 이 정의한 문법에 따라 값을 표현하는 식을 사용합니다.

JSP 는 웹 어플리케이션을 구현하는데 필요한 요청,응답,세션 등에 쉽게 접근할 수 있도록 request, response, session 등의 기본 객체를 제공하고 있습니다. JSP 는 EL 에서 사용할 수 있는 기본 객체도 제공하고 직접적으로 객체를 탐색할 수도 있습니다.

${pageScope.NAME}

위의 코드는 PAGE 영역에 저장되어 있는 NAME 이라는 속성값을 탐색합니다.

${name}

영역을 나타내는 EL 의 기본 객체를 사용하지 않고 이름만 지정하는 경우의 EL 은 4 개의 scope 를 차례대로 검색해서 속성이 존재하는지 확인한다. 그 검색 순서는

PAGE -> REQUEST -> SESSION -> APPLICAION

순으로 검색한다.

EL 의 표현식에는 수치 연산자, 비교 연산자, 논리 연산자, empty 연산자 등등 다양한 연산자를 사용할 수 있습니다. 먼저 수치 연산자를 예를 들면

${“10” + 1}

위와 같은 코드는 자바에 익숙한 개발자는 문자열 “101”이 될 것으로 예상하겠지만, 실제로 결과는 숫자 11 이다. EL 에서 수치 연산자는 정수 타입과 실수 타입에 대해서만 동작하고 자동으로 형변환을 하여 연산을 수행합니다.

비교 연산자는 == , != , < , > , <= , >= 등의 6 가지가 있습니다.

${someValue == “2005”}

위의 코드 처럼 표현식 안에서 연산자를 사용하여 값을 비교할 수 있다.

논리 연산자는 && , || , ! 3 가지의 연산자를 사용하고 사용법은 JAVA 와 같습니다.

empty 연산자는 검사할 객체가 텅 빈 객체인지를 검사하기 위해 사용합니다.

표준 태그 라이브러리(JSTL)

JSP 는 새로운 태그를 추가할 수 있는 기능을 제공하고 있는데, 이것이 바로 커스텀 태그 입니다. JSP 페이지에서 많이 사용되는 논리적인 판단, 반복 처리, 포맷 처리를 위한 커스텀 태그를 표준으로 만들어서 정의한 것이 있는데, 그것이 바로 JSTL(JSP Standard Tag Library)입니다.

코어 태그(core)

코어 태그 라이브러리는 변수 설정이나 if-else 와 같은 논리 처리에 사용되는 태그를 제공합니다.

코어 태그 라이브러리를 사용하려면 JSP 페이지에 taglib 디렉티브를 추가해야 합니다.

<%@ taglib prefix="c" uri=http://java.sun.com/jsp/jstl/core %>

기능 분류	태그	설명
변수 지원	set	JSP 에서 사용할 변수를 설정한다.
	remove	설정한 변수를 제거한다.
흐름 제어	if	조건에 따라 내부 코드를 수행한다.
	choose	다중 조건을 처리할 때 사용한다.
	forEach	컬렉션이나 Map 의 각 항목을 처리할 때 사용한다.
	forTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용한다.
URL 처리	import	URL 을 사용하여 다른 자원의 결과를 삽입한다.
	redirect	지정한 경로로 리다이렉트 한다.
	url	URL 을 재작성 한다.
기타 태그	catch	익셉션을 처리할 때 사용한다.
	out	JspWriter 에 내용을 출력한다.

1. <c:set> 태그

<c:set> 태그는 EL 변수의 값이나 EL 변수의 프로퍼티 값을 지정할 때 사용합니다.

```
<c:set var="변수명" value="값" [scope="영역"] />
<c:set var="변수명" [scope="영역"]>값</c:set>
```

- var : 값을 저장할 EL 변수의 이름을 지정한다.
- value : 변수의 값을 지정한다. 표현식, EL, 정적인 텍스트를 사용해서 값을 지정할 수 있다.
- scope : 변수를 저장할 영역을 지정한다. 값은 page, request, session, application 중 하나가 온다. 기본값은 page 이다.

2. .<c:remove> 태그

<c:remove> 태그는 <c:set>태그로 지정한 변수를 삭제할 때 사용한다.

```
<c:remove var="변수명" [scope="영역"] />
```

var 속성과 scope 속성은 <c:set> 태그의 두 속성과 동일한 의미를 갖습니다.

<c:remove>태그를 사용할 때 주의할 점은 삭제할 변수의 scope 를 지정하지 않으면 동일한 이름으로 저장된 모든 영역의 변수를 삭제합니다.

3. <c:if> 태그

<c:if> 태그는 Java 의 if 문법과 비슷한 기능을 합니다.

```
<c:if test="조건">
...
</c:if>
```

- test : true 혹은 false 에 해당하는 값이 온다. 이 조건 값이 true 이면 <c:if> 태그의 안쪽에 적힌 내용이 실행 된다.

4. <c:choose>, <c:when>, <c:otherwise> 태그

<c:choose> 태그는 자바의 if-else 문법과 비슷한 기능을 합니다.

```
<c:choose>
    <c:when test="${id == "aaa"}">
    ...
    </c:when>
    <c:when test="${id == "bbb"}>
    ...
    </c:when>
    <c:otherwise>
    ...
    </c:otherwise>
</c:choose>
```

<c:choose>태그는 다수의 <c:when>태그를 중첩해서 사용하는데 , 각각의 <c:when>태그는 test 속성이 true 일 때 내부 블록을 수행합니다. 실행된 <c:when>블록이 있을 경우에는 그 이후의 <c:when> 태그는 실행되지 않습니다.

만약 모든 <c:when>태그가 실행되지 않는다면<c:otherwise>태그가 실행 됩니다.

5. <c:forEach> 태그

<c:forEach>태그는 배열, 컬렉션 또는 Map 에 저장되어 있는 값들을 순차적으로 처리 할 때 사용됩니다.

```
<c:forEach var="변수" items="아이템">
    ${변수사용}
</c:forEach>
```

```
<c:forEach var="변수" begin="1" end="10">
    ${변수사용}
</c:forEach>
```

- items : 배열, 컬렉션 또는 Map 이 올 수 있다.
- begin : 반복의 시작시점
- end : 반복의 종료시점

6. <c:url>태그

<c:url>태그는 URL 을 생성해 주는 기능을 제공한다.

```
<c:url value="경로" />
```

- value : 절대 경로 혹은 상대 경로를 지정할 수 있다.

7. <c:out>태그

<c:out>태그는 JSP 페이지에 값을 출력할 때 사용 한다. <c:out>태그로 출력하는 값의 내용에 <,>,&,',” 등의 특수문자가 있을경우에는 기본적으로 HTML 코드로 변환하여 있는 그대로 출력된다.

```
<c:out value="값" />
```