# 16장 커스텀 태그 만들기

## 16.1 커스텀 태그

- JSP 액션 태그와 같은 태그를 추가할 수 있는 기능

### 16.1.1 커스텀 태그의 장점

- 재사용성 향상
- 쉽고 단순한 JSP 제작
- 가독성 향상

### 16.1.2 커스텀 태그의 종류

- JSP 1.2 스타일 커스텀 태그
- JSP 2.0 이상의 SimpleTag
- JSP 2.0 이상의 태그 파일 (구현이 쉬움)

## 16.2 예제를 위한 프로젝트 생성

- [File > New > Dynamic Web Project] 메뉴를 실행한다.
  - Project Name : chap16
- 프로젝트 생성 후 jstl-1.2.jar 파일을 WebContent/WEB-INF/lib 폴더에 복사한다.

## 16.3 태그 파일을 이용한 커스텀 태그 구현

### 16.3.1 태그 파일의 기본

- 태그 파일에서 사용할 수 있는 디렉티브

| 디렉티브 | 설명 |
|---|---|
| tag | JSP 페이지의 page 디렉티브와 동일하다. tag 디렉티브는 태그 파일의 정보를 명시한다. |
| taglib | JSP 페이지와 마찬가지로 태그 파일에서 사용할 태그 라이브러리를 명시할 때 사용한다. |
| include | JSP 페이지와 마찬가지로 태그 파일에 특정한 파일을 포함시킬 때 사용한다. 태그 파일에 포함되는 파일은 태그 파일에 알맞은 문법으로 작성되어야 한다. |
| attribute | 태그 파일이 커스텀 태그로 사용될 때 입력 받을 속성을 명시한다. |
| variable | EL 변수로 사용될 변수에 대한 정보를 지정한다. |

- tag 디렉티브의 주요 속성

| 속성 | 설명 |
|---|---|
| body-content | 몸체 내용의 종류를 지정 |
| dynamic-attributes | 동적 속성을 사용할 때, 속성의 <이름, 값>이 저장될 Map 객체를 page 범위의 속성에 저장할 때 사용할 이름을 명시 |
| import | page 디렉티브의 import 속성과 동일 (선택) |

| | |
|---|---|
| pageEncoding | page 디렉티브의 pageEncoding 속성과 동일 (선택) |
| isELIgnored | page 디렉티브의 isELIgnored 속성과 동일 (선택) |
| deferredSyntaxAllowedAsLiteral (2.1) | page 디렉티브의 deferredSyntaxAllowedAsLiteral 속성과 동일 (선택) |
| trimDirectiveWhitespaces (2.1) | page 디렉티브의 trimDirectiveWhitespaces 속성과 동일 (선택) |

### (1) 태그 파일의 위치와 태그 파일의 참조 방법

- 위치 및 확장자
  - WEB-INF/tags 디렉터리 및 하위 디렉터리
  - 확장자는 .tag 또는 .tagx

- JSP에서 태그 파일 참조
  - /WEB-INF/tags/util 디렉터리의 removeHtml.tag 파일을 <tf:removeHtml> 태그의 구현 코드로 사용

```
<%@ taglib prefix="tf" tagdir="/WEB-INF/tags/util" %>
 ...
<tf:removeHtml ...>...</tf:removeHtml>
```

### (2) 태그 파일에서 사용 가능한 기본 객체

- jspContext : pageContext가 제공하는 setAttribute(), getAttribute() 메서드를 그대로 제공하며, 각 속성과 관련된 작업을 처리한다.
- request : JSP 페이지의 request 기본 객체와 동일하다.
- response : JSP 페이지의 response 기본 객체와 동일하다.
- session : JSP 페이지의 session 기본 객체와 동일하다.
- application : JSP 페이지의 application 기본 객체와 동일하다.
- out : JSP 페이지의 out 기본 객체와 동일하다.

## 16.3.2 내용을 출력하는 단순 태그 파일 구현

[chap14/WebContent/WEB-INF/tags/now.tag]

```
01  <%@ tag body-content="empty" pageEncoding="utf-8" %>
02  <%@ tag import = "java.util.Calendar" %>
03  <%
04      Calendar cal = Calendar.getInstance();
05  %>
06  <%= cal.get(Calendar.YEAR) %>년
07  <%= cal.get(Calendar.MONTH)+1 %>월
08  <%= cal.get(Calendar.DATE) %>일
```

[chap14/WebContent/now.jsp]

```jsp
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ taglib prefix="tf" tagdir="/WEB-INF/tags" %>

<html>
<head><title>now</title></head>
<body>

오늘은 <b><tf:now /></b> 입니다.

</body>
</html>
```

## ~~16.3.3 태그 파일의 속성 설정 방법~~

■ JSTL의 <c:if> 태그나 <c:forEach> 태그를 보면 test, var, items 등의 속성을 이용해서 태그를 실행한는데 필요한 값을 전달받는다. 태그 파일도 속성을 이용해서 태그 파일을 실행하는데 필요한 값을 전달 받는다.

### (1) 속성값을 전달하는 기본 방식

■ attribute 디렉티브를 이용해서 지정한 속성은 다음과 같이 태그 파일의 스크립트 요소나 EL에서 변수처럼 사용할 수 있다.

■ attribute 디렉티브의 기본 사용법
- name : 속성 이름. 태그 파일에서 변수명으로 사용가능
- required : 필수 여부
- type : 타입 지정 (미지정시 java.lang.String_

```jsp
// 태그 파일의 스크립트 요소나 EL에서 변수처럼 사용
<%-- header.tag --%>
<%@ tag ... %>
<%@ attribute name="title" required="true" %>
...
<%= title %> 또는 ${title}

// JSP에서의 사용법
<tf:header title="<%= article.getTitle() %>" />
<tf:header title="${article.title}" />
<tf:header title="이 글의 제목" />
```

[chap16/WebContent/WEB-INF/tags/header.tag]

```jsp
<%@ tag body-content="empty" pageEncoding="utf-8" %>
<%@ tag trimDirectiveWhitespaces="true" %>
<%@ attribute name="title" required="true" %>
<%@ attribute name="level" type="java.lang.Integer" %>
<%
        String headStartTag = null;
        String headEndTag = null;
        if (level == null) {
                headStartTag = "<h1>";
                headEndTag = "</h1>";
        } else if (level == 1) {
                headStartTag = "<h1>";
```

```
13                      headEndTag = "</h1>";
14              } else if (level == 2) {
15                      headStartTag = "<h2>";
16                      headEndTag = "</h2>";
17              } else if (level == 3) {
18                      headStartTag = "<h3>";
19                      headEndTag = "</h3>";
20              }
21      %>
22      <%= headStartTag %>
23      ${title}
24      <%= headEndTag %>
```

[chap16/WebContent/use_header.jsp]

```
01      <%@ page contentType = "text/html; charset=utf-8" %>
02      <%@ taglib prefix="tf" tagdir="/WEB-INF/tags" %>
03
04      <html>
05      <head><title>제목 출력</title></head>
06      <body>
07
08      <tf:header title="텍스트 제목" level="2"/>
09
10      <tf:header title="${'EL 제목'}" level="3"/>
11
12      <tf:header title='<%= "표현식 제목" %>' />
13
14      </body>
15      </html>
```

### (2) <jsp:attribute> 액션 태그를 이용한 속성값 전달

- attribute 디렉티브의 fragment 속성값이 true일 경우, JSP는 속성에 값을 전달할 때 <jsp:attribute> 액션 태그를 사용해야 한다.

```
// 태그 파일
<%-- header.tag --%>
<%@ attribute name="title" fragment="true" %>

// 첫 번째, <jsp:attribute>의 몸체 내용을 그대로 처리하여 출력
<jsp:invoke fragment="title" />

// 두 번째, <jsp:attribute>의 몸체 내용을 처리한 결과를 지정한 영역의 속성에 저장
<jsp:invoke fragment="title" var="rs" scope="page" />
${pageScope.rs}

// JSP 파일
<tf:header>
    <jsp:attribute name="title">${article.title</jsp:attribute>
</tf:header>
```

### (3) 동적 속성 전달

- attribute 디렉티브로 지정하지 않은 속성을 Map에 저장

■ tag 디렉티브의 dynamic-attributes 속성에 동적 속성을 저장할 변수명 지정 (변수 타입은 Map)

```
<%@ tag dynamic-attributes="dynamicMap" %>
...
${dynamicMap.attrName} .. <!-- attrName 속성값 -->
```

## ~~16.3.4 몸체 내용 처리~~

```
// 방법1 - 태그 몸체
<tf:someTagFile attr1="속성값">
여기에 몸체 내용을 입력한다.
</tf:someTagFile>

// 방법 2 - <jsp:body> 액션 태그 사용
<tf:someTagFile attr1="속성값">
    <jsp:attribute name="attr2">value</jsp:attribute>
    <jsp:body>
    여기에 몸체 내용을 입력한다.
    </jsp:body>
</tf:someTagFile>
```

### (1) EL과 태그를 처리한 몸체 내용 사용하기

```
<%@ tag body-content="scriptless" %>
<!-- 몸체로 전달 받은 내용을 사용 -->
<jsp:doBody />
```

### (2) 몸체 내용 자체를 데이터로 사용하기

```
<%@ tag body-content="tagdependent" pageEncoding="euc-kr" %>
…
<jsp:doBody var="bodyText" />
```

### (3) 몸체 내용을 반복적으로 사용하기

■ 액션 태그를 반복적으로 실행하면 된다.

## ~~16.3.5 변수의 생성~~

### (1) variable 디렉티브와 name-given을 이용한 변수 추가

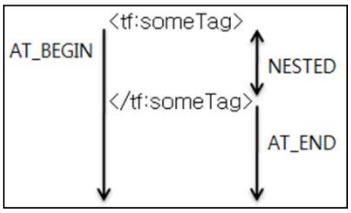
■ variable 디렉티브를 사용해서 태그 파일을 사용하는 JSP/태그 파일에서 사용할 EL 변수 추가

■ variable 디렉티브의 기본 문법
- name-given 속성 및 scope 속성에 따른 EL 변수 특징

```
태그 파일
<%@ variable name-given="sum" scope="NESTED" %>
...

태그를 호출하는 페이지
<%@ taglib prefix="tf" tagdir="/WEB-INF/tags" %>

<tf:some begin="1" end="10">
합계 - ${sum}
</tf:some>
```




### (2) variable 디렉티브와 name-from-attribute 속성을 이용한 변수 생성

■ 특정 속성의 값을 생성할 변수명으로 사용

■ name-from-attribute 속성의 사용법
- name-from-attribute 속성에 명시한 속성의 제약 조건
  - rtexprvalue 속성의 값이 false
  - required 속성이 true
  - type 속성의 값은 java.lang.String
- alias 속성 : 태그 파일에서 생성할 변수에 접근할 때 사용할 EL 변수명

```
<%@ attribute name="var" rtexprvalue="false"
              required="true" type="java.lang.String" %>
<%@ variable alias="localName" name-from-attribute="var" scope="영역" %>
...
<c:set var="localName" value="..." />
```

### (3) 변수의 동기화 처리

| 구분 | AT_BEGIN | NESTED | AT_END |
|---|---|---|---|
| 태그 시작 | 아무것도 안 함 | EL 변수 저장 | 아무것도 안 함 |
| <jsp:doBody> 실행 전 | 태그→페이지 복사 | 태그→페이지 복사 | 아무것도 안 함 |
| 태그 끝 | 태그→페이지 복사 | EL 변수 복구 | 태그→페이지 복사 |