

커스텀 태그 만들기

TOC

- 커스텀 태그란
- 태그 파일 기본
- 태그 파일의 기본 객체
- 속성 설정
- 몸체 내용 처리
- 변수 생성

커스텀 태그(Custom Tag)

- JSP 액션 태그와 같은 태그를 추가할 수 있는 기능
- 원하는 목적에 맞게 작성된 태그
- 커스텀 태그의 장점
 - 재사용성 향상
 - 쉽고 단순한 JSP 제작
 - 가독성 향상
- 커스텀 태그의 종류
 - JSP 1.2 스타일 커스텀 태그
 - JSP 2.0 이상의 SimpleTag
 - JSP 2.0 이상의 태그 파일 (구현이 쉬움)

태그 파일(tag file)

- JSP 문법을 사용해서 커스텀 태그로 동작할 수 있도록 만들어진 소스 코드
- 서블릿보다 JSP가 작성하기 쉽듯이, 클래식 태그나 SimpleTag보다 태그 파일이 작성하기 쉬움
- JSP와 동일한 방식으로 태그 파일 작성

태그 파일의 위치 및 참조

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

- JSP에서 태그 파일 참조

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

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

태그 파일에서 사용할 수 있는 디렉티브

디렉티브	설 명
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

속 성	설 명
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	page 디렉티브의 deferredSyntaxAllowedAsLiteral 속성과 동일 (선택)
trimDirectiveWhitespaces	page 디렉티브의 trimDirectiveWhitespaces 속성과 동일 (선택)

태그 파일의 기본 객체

- `jspContext` - `pageContext`가 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.

attribute 디렉티브 - 기본

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

```
<%-- 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" />
```

<jsp:attribute> 액션 태그 이용 속성 값 전달

- 태그 파일의 attribute 디렉티브의 fragment 속성 값이 true일 경우 JSP에서는 속성에 값을 전달할 때 <jsp:attribute> 액션 태그를 사용해야 함. 즉

```
<%-- header.tag --%>
<%@ attribute name="title" fragment="true" %>
...
// 첫번째, <jsp:attribute>의 몸체 내용을 그대로 처리하여 출력
<jsp:invoke fragment="title" />

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

<%-- JSP --%>
<@ taglib prefix="tf" tagdir="/WEB-INF/tags" %>
<tf:header>
    <jsp:attribute name="title">${article.title}</jsp:attribute>
</tf:header>
```

- 몸체 내용에 텍스트, EL, <jsp:include> 액션 태그 사용 가능.
- 단, 스크립트 코드(표현식과 스크립트릿)는 사용 불가.

동적 속성

- 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 -->
```

태그 파일 이용 커스텀 태그에 몸체 내용 전달하기

- 방법1 - 태그 몸체

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

- 방법2 - <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>
```

– 태그 파일 이용 커스텀 태그의 몸체에는 표현식 사용 불가

태그 파일에서 몸체 내용 사용하기

- EL 및 태그가 처리된 몸체 내용 사용하기

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

- 몸체 내용을 데이터로 사용하기

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

variable 디렉티브와 name-given을 이용한 변수 추가

- variable 디렉티브를 사용해서 태그 파일을 사용하는 JSP/태그 파일에서 사용할 EL 변수 추가
- variable 디렉티브의 기본 문법

```
<%@ variable name-given="EL변수이름"  
      variable-class="변수타입"  
      scope="변수범위" />
```

– name-given 속성 및 scope 속성에 따른 EL 변수 특징

태그 파일

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

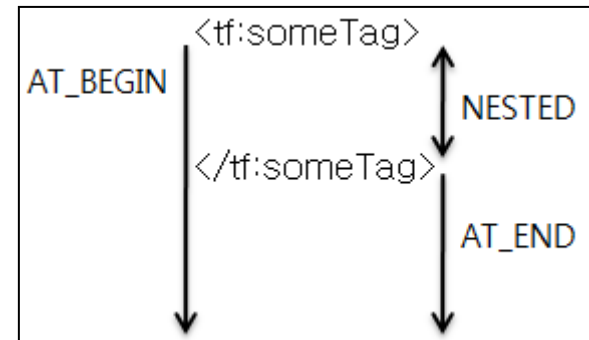
태그를 호출하는 페이지

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

```
<tf:some begin="1" end="10">
```

```
합계 - ${sum} ←
```

```
</tf:some>
```



name-from-attribute 속성을 이용한 변수 생성

- 특정 속성의 값을 생성할 변수명으로 사용
- name-from-attribute 속성의 사용법

```
<%@ 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="..." />
```

- 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

변수 동기화 처리

	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