



Chapter19

커스텀 태그(Custom Tag)



Contents

학습목표

- ❖ 커스텀 태그의 문법을 이해한다.
- ❖ 커스텀 태그의 세부요소를 이해한다.
- ❖ 필요한 커스텀 태그를 직접 만들어 본다.

학습내용

- ❖ 커스텀 태그의 이해
- ❖ 커스텀 태그의 세부요소
- ❖ 커스텀 태그 예제



커스텀 태그의 이해

❖ 커스텀 태그란?

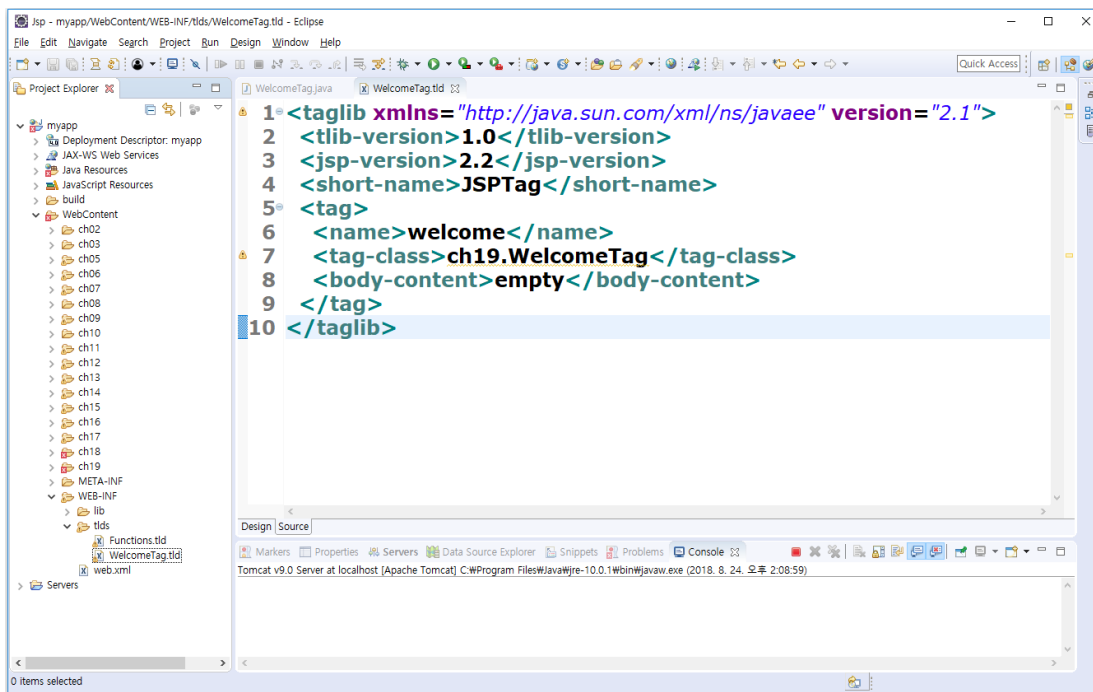
- 사용자 태그란 직접 프로그래머가 자신만의 태그를 만들어 사용할 수 있는 기술입니다. 사용자 태그의 필요성은 JSP 페이지 안에 있는 많은 태그들(스크립트릿에 포함된 많은 자바코드들)이 HTML 태그와 같이 있을 때 페이지 전체가 복잡해진다는 사실에서 출발합니다.
- JSP 페이지 속에 많은 자바코드들이 HTML 코드와 혼재되어 있을 때 일반적으로 프로그래머 자신이 보기에 복잡해지겠지만 페이지 디자이너들이 HTML과 JSP 코드들을 구별해서 디자인하기는 더욱 쉽지 않다는 것입니다.
- 사용자 태그(Custom Tag)라는 기술이 JSP1.1 기술에서부터 시작하여 활성화 되고 있는 것입니다. 이렇게 사용자가 만든 태그들의 집합체를 '태그 라이브러리'라고 표현합니다.



커스텀 태그의 세부요소

❖ 커스텀 태그 세부요소

- 사용자 태그를 만들기 위해서는 다음과 같은 3개의 파일을 작성해야한다.
 1. 태그 핸들러(Tag Handler-자바코드) 작성
 2. 태그 라이브러리 설명자 파일(Tag Library Descriptor-TLD) 작성
 3. JSP 페이지 내에서 TLD 지정과 사용자 태그 호출



```
1 <taglib xmlns="http://java.sun.com/xml/ns/javaee" version="2.1">
2   <tlib-version>1.0</tlib-version>
3   <jsp-version>2.2</jsp-version>
4   <short-name>JSPTag</short-name>
5   <tag>
6     <name>welcome</name>
7     <tag-class>ch19.WelcomeTag</tag-class>
8     <body-content>empty</body-content>
9   </tag>
10 </taglib>
```




커스텀 태그 세부요소

❖ “Welcome to My Custom Tag” 사용자 태그 만들기

- 실습소스(java) : <source/ch19/WelcomeTag.java>

❖ Tag Library Descriptor(TLD) 만들기

- 실습소스(tld) : <source/ch19/WelcomeTag.tld>

❖ “Welcome to My Custom Tag” JSP 만들기

- 실행화면 : <http://jspstudy.co.kr/myapp/ch19/welcomeTag.jsp>
- 실행소스 : <source/ch19/WelcomeTag.jsp>





커스텀 태그의 세부요소

❖ 커스텀 태그 세부요소

- 태그 라이브러리 설명자 파일(Tag Library Descriptor-TLD) 작성1

▶ 보기 : TLD 파일의 구성1

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee" version="2.1">
  <tlib-version>1.0</tlib-version>
  <jsp-version>2.2</jsp-version>
  <short-name>JSPTag</short-name>
  <tag>....</tag>
  <tag>....</tag>
  ....
</taglib>
```

태그	설명
tlib-version	태그 라이브러리 버전을 나타냅니다. 마치 개발자가 자신이 개발한 프로그램에 버전을 표시하는 것과 같습니다.
jsp-version	현재 사용되고 있는 JSP 버전을 말합니다. 이것은 현재 개발한 사용자 태그는 JSP 2.2문법(규약, 약속)에 따르고 있음을 말해 줍니다.
short-name	태그 라이브러리의 간단한 이름을 지정해 줍니다. 이 이름은 JSP 개발도구나 JSP 컨테이너(ex-Tomcat)등에서 태그 라이브러리의 이름으로 사용되는 부분입니다. 또한 jsp 페이지에서 prefix의 값으로 사용될 수 있습니다. 이 이름을 작성할 때에 공백은 허용되지 않고 첫 글자에 숫자 혹은 '_'(언더바)는 사용 할 수 없습니다.

▲ [표 19-1] TLD 파일을 구성하는 태그



커스텀 태그의 세부요소

❖ 커스텀 태그 세부요소

- 태그 라이브러리 설명자 파일(Tag Library Descriptor-TLD) 작성2

▶ 보기 : TLD 파일의 구성2

```
<taglib>
....
  <tag>
    <name>welcome</name>
    <tag-class>ch19.WelcomeTag</tag-class>
    <body-content>empty</body-content>
  </tag>
....
</taglib>
```

태그	설명
name	사용자 태그의 이름을 지정합니다.
tag-class	사용자 태그의 처리를 담당해 줄 태그 핸들러를 지정합니다.
body-content	사용자 태그의 몸체가 있는지 없는지의 여부를 지정합니다.

▲ [표 19-2] <tag> 구성하는 하위 태그



커스텀 태그의 세부요소

❖ 커스텀 태그 세부요소

- 태그 핸들러(Tag Handler-자바코드) 작성
 - 태그 핸들러란 사용자 태그를 직접적으로 담당하여 처리하고 결과물을 생성해 주는 클래스 파일을 말합니다.
 - javax.servlet.jsp.tagext.Tag, JspTag, SimpleTag 인터페이스 구현

클래스 이름	설명	JSP version
TagSupport	몸체 내용을 처리하지 않은 경우	JSP 1.2
BodyTagSupport	몸체 내용을 처리하는 경우	JSP 1.2
SimpleTagSupport	몸체 내용 처리 여부에 관계없이 JSP 2.0 이상에서 사용 할 수 있는 핸들러 클래스	JSP 2.0

▲ [표 19-3] 태그 핸들러 만들기 위한 클래스

클래스 이름	설명
몸체가 없는 경우	<tag:welcome></tag:welcome> 혹은 <tag:welcome/>
몸체가 있는 경우	<tag:welcome>JSPStudy</tag:welcome>

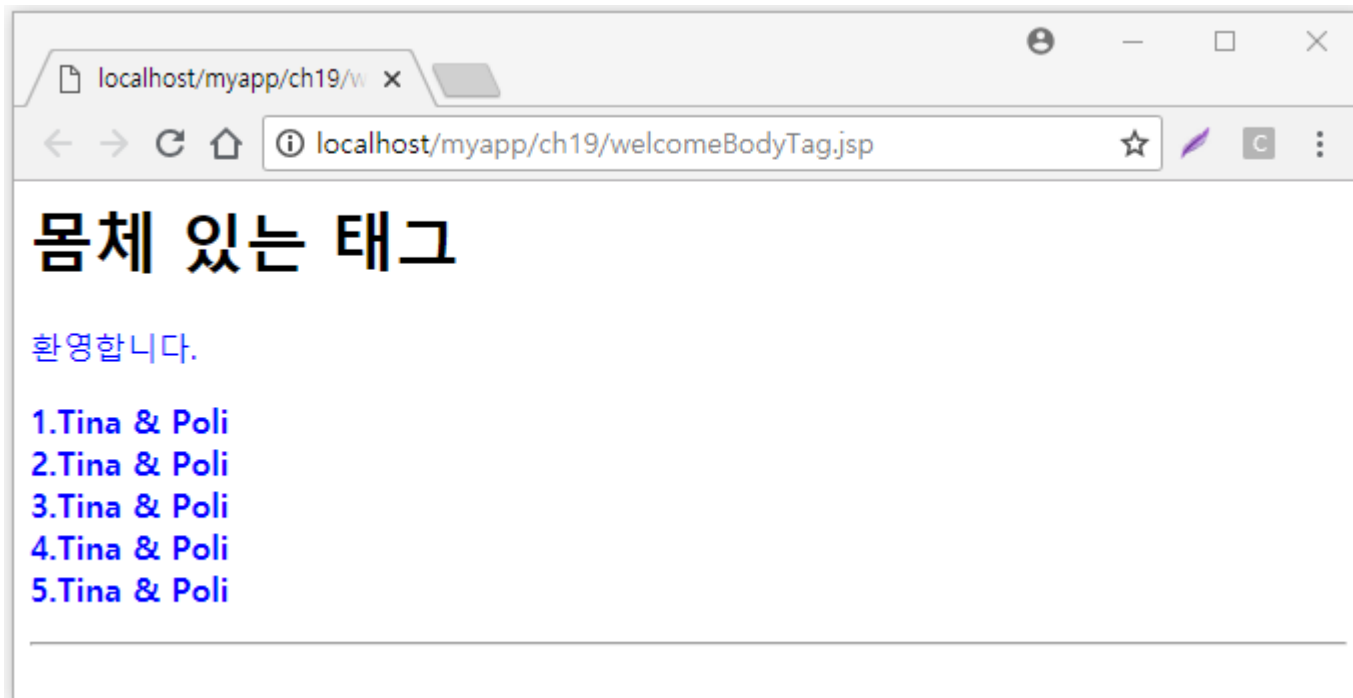
▲ [표 19-4] 몸체의 구분



커스텀 태그의 세부요소

❖ 사용자 태그의 몸체를 여러 번 출력하는 예제

- 실행화면 : <http://jspstudy.co.kr/myapp/ch19/welcomeBodyTag.jsp>
- 실습소스(java) : <source/ch19/WelcomeBodyTag.java>
- 실습소스(tld) : <source/ch19/WelcomeBodyTag.tld>
- 실행소스(jsp) : <source/ch19/welcomeBodyTag.jsp>





속성값이 있는 커스텀 태그

❖ 속성값이 있는 사용자 태그 작성

```
<examples:tagName att1="..." att2="..."/>
<examples:tagName att1="..." att2="...">몸체 내용</examples:tagName>
```

```
<attribute>
  <name>from</name>
  <type>java.lang.Integer</type>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>to</name>
  <type>java.lang.Integer</type>
  <rtexprvalue>true</rtexprvalue>
</attribute>
```

속성을 위해 추가한 태그입니다.

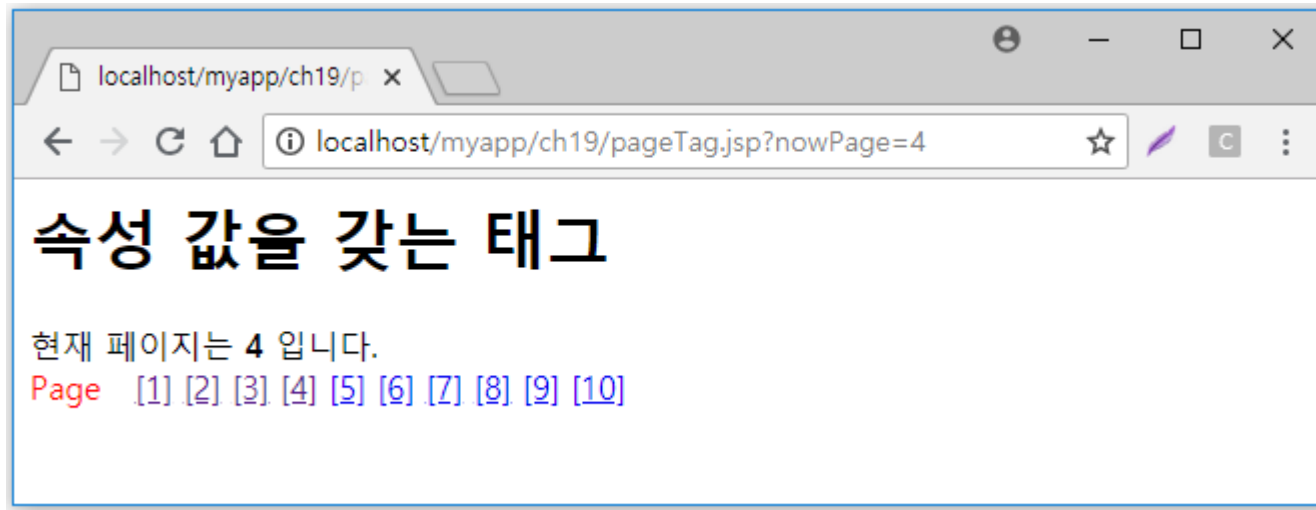
name	사용자 태그에서 사용된 속성의 이름을 지정해 줍니다. 이 태그는 반드시 지정해야 하는 태그입니다.
type	사용자 태그에서 사용된 속성의 data 타입을 지정해 줍니다. java.lang.String, java.lang.Integer 이러한 값들이 올 수 있습니다.



속성값이 있는 커스텀 태그

❖ 속성값이 있는 사용자 태그 예제

- 실행화면 : <http://jspstudy.co.kr/myapp/ch19/pageTag.jsp>
- 실습소스(java) : <source/ch19/PageTag.java>
- 실습소스(tld) : <source/ch19/PageTag.tld>
- 실행소스(jsp) : <source/ch19/pageTag.jsp>





스크립트 변수 생성하기

❖ 스크립트 변수 생성하기

■ 보기 : 스크립트 변수 사용

```
<jsp:useBean id = "db" class = "package.DBConnection">
<%
    Connection conn = db.getConnection();
%>
```

위 보기에서 보시는 것처럼 빈 태그에서 객체를 생성해서 db라는 스크립트 변수 이름(객체명)으로 접근해서 객체가 가지고 있는 기능을 사용할 수 있습니다. 여기서는 DBConnection 클래스의 객체 db를 생성하여 Connection을 연결하는 모습입니다.

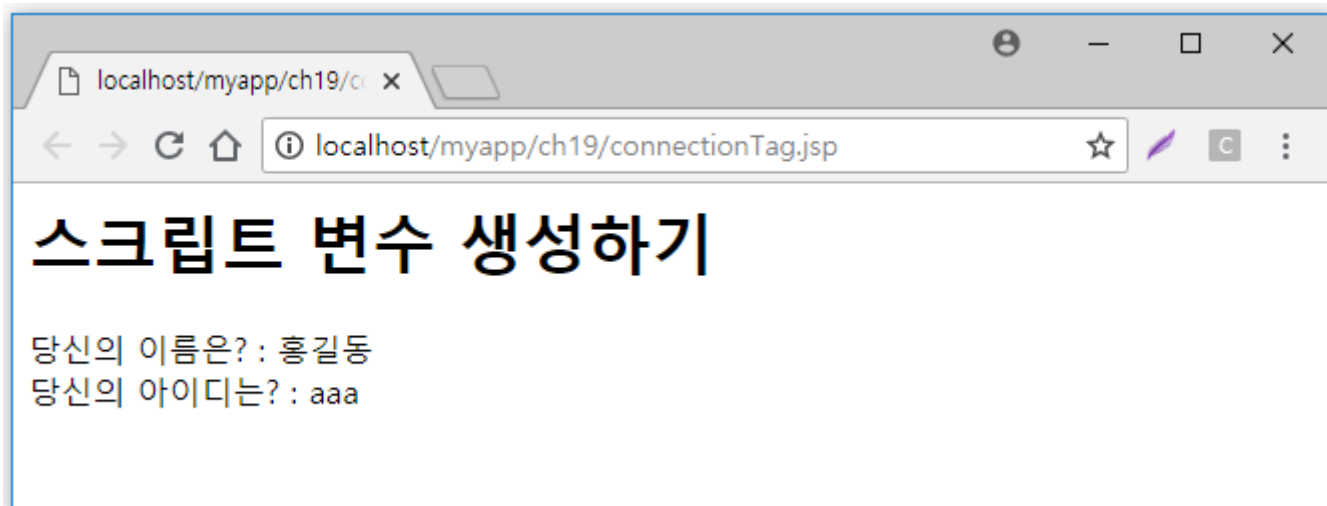
- 스크립트 변수를 생성하여 사용하기 위해서는 TLD파일과 태그 핸들러 외에도 두가지 더 필요하게 됩니다.
 - 첫번째 : 스크립트(변수)를 생성하기 위한 클래스
 - 두번째 : 스크립트 변수의 검사를 수행하는 Helper 클래스



스크립트 변수 생성하기

❖ 속성값이 있는 사용자 태그 예제

- 실행화면 : <http://jspstudy.co.kr/myapp/ch19/connectionTag.jsp>
- 실습소스(Helper-java) : <source/ch19/ConnectionTagTEI.java>
- 실습소스(스크립트-java) : <source/ch19/DBConnection.java>
- 실습소스(핸들러-java) : <source/ch19/ConnectionTag.java>
- 실습소스(tld) : <source/ch19/ConnectionTag.tld>
- 실행소스(jsp) : <source/ch19/connectionTag.jsp>





Thank You !