

```

1 Custom Tag?
2     1. 사용자가 정의해서 사용하는 Tag
3     2. JSTL 에서 지원하지 못하는 태그 형식을 사용할 때
4     3. JSP 1.1 에서 도입
5
6
7 1. 장점
8     1)재사용성
9     2)배포가능
10    3)가독성
11    4)의존성
12    5)디자이너와 개발자의 역할 분담.
13
14
15 2. 종류
16     1) JSP 1.2 스타일로 구현한 커스텀 태그 --> 자바 이해 필수 --> Classic custom Tag
17     2) JSP 2.0 또는 그 이상의 버전에서 구현한 커스텀 태그 --> 자바 이해 필수
18     3) JSP 2.0 또는 그 이상의 버전에서 Tag 파일을 이용해서 구현한 커스텀 태그
19
20
21 3. JSP 1.2 스타일로 구현한 커스텀 태그
22     1) 작성순서
23         -Java Class 파일 생성
24         -TLD(Tag Library Descriptor) 파일 생성
25         -Custom Tag 생성
26
27     2)Lab
28         -customtagdemo.jsp
29             <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
30             <%@ taglib prefix="aaa" uri="/WEB-INF/tlds/ccc.tld" %>
31             <!DOCTYPE htm>
32             <html lang="ko">
33             <head>
34             <meta charset="UTF-8" />
35             <title>Insert title here</title>
36             </head>
37             <body>
38                 <aaa:bbb /> <br />
39                 <font size='4' color='blue'>Welcome Custom Tag</font>
40             </body>
41             </html>
42
43         -ccc.tld
44             <?xml version="1.0" encoding="UTF-8" ?>
45             <taglib xmlns="http://java.sun.com/xml/ns/j2ee"
46                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
47                 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
48                 version="2.0">
49             <description>JSP Custom Tag 1.2 연습 태그</description>
50             <tlib-version>1.1</tlib-version>    <-----필수
51             <short-name>aaa</short-name>    <-----필수
52             <tag>
53                 <description>빨간글씨로 HelloWorld 출력하기</description>
54                 <name>bbb</name>    <-----필수
55                 <tag-class>kr.co.javaexpert.libs.HelloWorld</tag-class>    <-----필수
56                 <body-content>empty</body-content>    <-----필수
57                 <---body 가 존재하지 않으면 empty, 있으면 JSP 로 입력
58             </tag>
59             </taglib>
60
61         -HelloWorld.java
62             package com.example.libs;
63
64             import java.io.IOException;
65             import javax.servlet.jsp.tagext.TagSupport;
66             import javax.servlet.jsp.JspTagException;
67             import javax.servlet.jsp.JspWriter;

```

```

68
69     public class HelloWorld extends TagSupport{
70         @Override
71         public int doStartTag() throws JspTagException{
72             JspWriter out = pageContext.getOut();
73             try {
74                 out.print("<font size='7' color='red'>");
75                 out.println("Hello, World</font>");
76             } catch (Exception e) {
77                 throw new JspTagException(e.getMessage());
78             }
79             return this.SKIP_BODY;
80         }
81         @Override
82         public int doEndTag() throws JspTagException{
83             return this.EVAL_PAGE;
84         }
85     }
86
87

```

3) javax.servlet.jsp.tagext.Tag interface or javax.servlet.jsp.tagext.TagSupport class

```

89 -int EVAL_BODY_INCLUDE
90 -int SKIP_BODY
91 -int EVAL_PAGE
92 -int SKIP_PAGE
93 -public int doStartTag() throws JspException
94     --시작 태그를 만날 때 실행
95 -public int doEndTag() throws JspException
96     --끝 태그를 만날 때 실행
97 -public void setPageContext(PageContext pageContext)
98     --전달 받은 pageContext 객체를 저장할 때 사용
99
100

```

#### 4. Tag Style

1)<br /> : body 도 없고, 속성도 없는 태그

```

103 -date.jsp
104     <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
105     <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mydate.tld" %>
106     <body>
107         Current time is <mylib:date />.
108     </body>
109

```

-mydate.tld

```

111     <?xml version="1.0" encoding="UTF-8" ?>
112     <taglib xmlns="http://java.sun.com/xml/ns/j2ee"
113         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
114         xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
115             http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
116         version="2.0">
117         <tlib-version>1.0</tlib-version>
118         <short-name>mylib</short-name>
119         <tag>
120             <name>date</name>
121             <tag-class>kr.co.javaexpert.libs.MyDate</tag-class>
122             <body-content>empty</body-content>
123         </tag>
124     </taglib>
125

```

-MyDate.java

```

127     package com.example.libs;
128
129     import java.util.Date;
130
131     import javax.servlet.ServletException;
132     import javax.servlet.jsp.JspTagException;
133     import javax.servlet.jsp.JspWriter;
134     import javax.servlet.jsp.tagext.TagSupport;

```

```

135
136 public class MyDate extends TagSupport {
137     @Override
138     public int doStartTag() throws JspTagException{
139         JspWriter out = pageContext.getOut();
140         ServletRequest request = pageContext.getRequest();
141         try {
142             out.println(new Date() + "<br />");
143             out.println("Client IP : " + request.getRemoteAddr());
144         } catch (Exception e) {
145             throw new JspTagException(e.getMessage());
146         }
147         return this.SKIP_BODY;
148     }
149     @Override
150     public int doEndTag() throws JspTagException{
151         return this.EVAL_PAGE;
152     }
153 }
154

```

2)<hr size='2' color='red' width='50%' align='left' /> :

-속성은 있지만, body 가 없는 태그

-date2.jsp

```

158 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
159 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mylib.tld" %>
160 <%
161     String type = request.getParameter("type");
162     if(type == null || type.length() == 0) type = "0";
163 %>
164 <!DOCTYPE html>
165 <html lang="ko">
166 <head>
167 <meta charset="UTF-8">
168 <title>Custom Tag Demo</title>
169 </head>
170 <body>
171 <h1>원하시는 스타일을 지정해 주세요</h1>
172 <form method="get">
173     <input type="radio" name="type" value="0"/> 한국 <br />
174     <input type="radio" name="type" value="1"/> 일본 <br />
175     <input type="radio" name="type" value="2"/> 미국 <br />
176     <input type="radio" name="type" value="3"/> 영국 <br />
177     <input type="radio" name="type" value="4"/> 중국 <br />
178     <input type="radio" name="type" value="5"/> 프랑스 <br />
179     <input type="submit" value="전송" />
180 </form>
181 <hr color='red' size='3' width='90%' />
182 Now is <mylib:date2 type="<%=type %>" />
183 </body>
184 </html>
185

```

-mylib.tld

```

187 <?xml version="1.0" encoding="UTF-8" ?>
188 <taglib xmlns="http://java.sun.com/xml/ns/j2ee"
189     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
190     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
191     version="2.0">
192     <tlib-version>2.0</tlib-version>
193     <short-name>mylib</short-name>
194     <tag>
195         <name>date</name>
196         <tag-class>kr.co.javaexpert.libs.MyDate</tag-class>
197         <body-content>empty</body-content>
198     </tag>
199     <tag>
200         <name>date2</name>
201         <tag-class>kr.co.javaexpert.libs.MyDate2</tag-class>

```

```

202         <body-content>empty</body-content>
203     <attribute>
204         <name>type</name>
205         <required>>false</required>
206         <rtexprvalue>>true</rtexprvalue>
207     </attribute>
208 </tag>
209 <tag>
210     <name>header</name>
211     <tag-class>kr.co.javaexpert.libs.MyHeader</tag-class>
212     <body-content>empty</body-content>
213     <attribute>
214         <name>name</name>
215         <required>true</required>
216         <rtexprvalue>true</rtexprvalue>
217     </attribute>
218 </tag>
219 <tag>
220     <name>content</name>
221     <tag-class>kr.co.javaexpert.libs.MyContent</tag-class>
222     <body-content>JSP</body-content>
223 </tag>
224
225 </taglib>
226
227 -MyDate2.java
228     package com.example.libs;
229
230     import java.text.DateFormat;
231     import java.util.Date;
232     import java.util.Locale;
233
234     import javax.servlet.jsp.JspException;
235     import javax.servlet.jsp.JspWriter;
236     import javax.servlet.jsp.tagext.TagSupport;
237
238     public class MyDate2 extends TagSupport {
239         private String type;
240         public void setType(String type){
241             this.type = type;
242         }
243         public int doStartTag() throws JspException{
244             JspWriter out = pageContext.getOut();
245             DateFormat df = null;
246             try{
247                 Date current = new Date();
248                 Locale locale = null;
249                 if(type == null) type="0";
250                 switch(type){
251                     case "0" : locale = Locale.KOREA; break;
252                     case "1" : locale = Locale.JAPAN; break;
253                     case "2" : locale = Locale.US; break;
254                     case "3" : locale = Locale.UK; break;
255                     case "4" : locale = Locale.CHINA; break;
256                     case "5" : locale = Locale.FRENCH; break;
257                 }
258                 df = DateFormat.getDateInstance(
259                     DateFormat.FULL, DateFormat.FULL, locale);
260                 out.println(df.format(current));
261             }catch(Exception ex){
262                 throw new JspException(ex.getMessage());
263             }
264             return SKIP_BODY;
265         }
266         public int doEndTag() throws JspException{
267             type = "0";
268             return EVAL_PAGE;

```

```

269     }
270 }
271
272 -header.jsp
273 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
274 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mylib.tld" %>
275 <body>
276     <h2>Request Headers</h2>
277     1. user-agent : <mylib:header name="user-agent" /> <br />
278     2. accept : <mylib:header name="accept" /> <br />
279     3. cookie : <mylib:header name="cookie" /> <br />
280 </body>
281
282 -MyHeader.java
283 package com.example.libs;
284
285 import java.util.Enumeration;
286 import javax.servlet.http.HttpServletRequest;
287 import javax.servlet.jsp.JspTagException;
288 import javax.servlet.jsp.JspWriter;
289 import javax.servlet.jsp.tagext.TagSupport;
290
291 public class MyHeader extends TagSupport {
292     private String name;
293     public void setName(String name){
294         this.name = name;
295     }
296     @Override
297     public int doStartTag() throws JspTagException{
298         JspWriter out = pageContext.getOut();
299         HttpServletRequest req = (HttpServletRequest)pageContext.getRequest();
300         try {
301             out.print(req.getHeader(this.name));
302         } catch (Exception e) {
303             throw new JspTagException(e.getMessage());
304         }finally{
305             return this.SKIP_BODY;
306         }
307     }
308     @Override
309     public int doEndTag() throws JspTagException{
310         return this.EVAL_PAGE;
311     }
312 }
313
314 3)<font size='5' color='red' face='Arial'>Hello</font> :
315 -속성도 있고, body 도 있는 태그
316 -content.jsp
317 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
318 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mylib.tld" %>
319 <body>
320     <h3>Tag Body 연습</h3>
321     <mylib:content>
322         구성모
323     </mylib:content>
324 </body>
325
326 -MyContent.java
327 package com.example.libs;
328
329 import javax.servlet.jsp.JspTagException;
330 import javax.servlet.jsp.JspWriter;
331 import javax.servlet.jsp.tagext.TagSupport;
332
333 public class MyContent extends TagSupport {
334     @Override
335     public int doStartTag() throws JspTagException{

```

```

336         try {
337             pageContext.getOut().print("반갑습니다.");
338         } catch (Exception e) {
339             throw new JspTagException(e.getMessage());
340         }finally{
341             return this.EVAL_BODY_INCLUDE;
342         }
343     }
344     @Override
345     public int doEndTag() throws JspTagException{
346         try{
347             pageContext.getOut().println("님! ^-^;");
348         }catch(Exception ex){
349             throw new JspTagException(ex.getMessage());
350         }finally{
351             return this.EVAL_PAGE;
352         }
353     }
354 }

```

#### 4) web.xml 코드 (<--JSP 2.x 이상에서는 필요없음)

```

<taglib>
  <taglib-uri>/WEB-INF/tlds/ccs.tld</taglib-uri>
  <taglib-location></taglib-location>
</taglib>

```

#### 5. JSP 2.0 또는 그 이상의 버전에서 구현한 커스텀 태그

- 1)JSP 1.1 과 JSP 1.2의 커스텀 태그와 태그 핸들러가 작성하기 너무 어려웠다.
- 2)그래서 JSP 2.0 에서 javax.servlet.jsp.tagext의 SimpleTag interface 추가
- 3)SimpleTag 를 구현한 태그 핸들러는 Simple Tag Handler 라 부른다.
- 4)기존 Tag, IterationTag, BodyTag 를 구현한 태그 핸들러는 클래식 태그 핸들러라고 부른다.
- 5)작성하기 쉽고, 생명주기도 단순하다.
- 6)태그 실행과 관련된 메소드는 doTag 하나뿐
- 7)이 메소드는 태그를 실행할 때 한번 호출
- 8)심플 태그 핸들러의 생명주기
  - JSP Container 는 인자 없는 생성자를 호출해서 인스턴스를 생성, 따라서 반드시 인자 없는 생성자를 가져야 함.
  - JSP Container 는 setJspContext 메소드를 호출하고, JspContext 객체를 인자로 넘긴다. JspContext 의 메소드 중 가장 중요한 메소드는 getOut 으로, 이 메소드는 클라이언트에 응답을 출력하기 위한 JspWriter 를 반환한다. setJspContext 의 시그니처는 다음과 같다.

```

public void setJspContext(JspContext jspContext)

```

-custom tag 가 다른 태그 안에 포함될 경우 JSP 컨테이너는 setParent 메소드를 호출한다.

```

public void setParent(JspTag parent)

```

-Jsp Container 는 태그의 각 속성별 setter 메소드를 호출한다.

-body 가 있을 경우 JSP Container 는 SimpleTag interface의 setJspBody 메소드를 호출해서 바디의 내용을 담은 JspFrament 를 전달한다. 바디가 없다면 Jsp Container 는 이 메소드를 호출하지 않는다.

-Jsp Container 는 doTag 메소드를 호출한다. doTag 메소드가 리턴할 때 모든 변수는 동기화된 상태이다.

- 9)Tag interface를 구현한 TagSupport Class가 아니라, SimpleTag interface를 구현한 SimpleTagSupport class를 사용한다.

#### 10) Java class 수정

```

public class HelloWorld extends TagSupport{
    @Override
    public int doStartTag() throws JspTagException{
        JspWriter out = pageContext.getOut();
        try {
            out.print("<font size='7' color='red'>");
            out.println("Hello, World</font>");
        } catch (Exception e) {
            throw new JspTagException(e.getMessage());
        }
        return this.SKIP_BODY;
    }
    @Override
    public int doEndTag() throws JspTagException{
        return this.EVAL_PAGE;
    }
}

```

```

400     }
401 }
402
403 -HelloWorld.java
404     public class HelloWorld extends SimpleTagSupport{
405         @Override
406         public void doTag() throws JspException, java.io.IOException{
407             JspWriter out = pageContext.getOut();
408             try {
409                 out.print("<font size='7' color='red'>");
410                 out.println("Hello, World</font>");
411             } catch (Exception e) {
412                 throw new JspTagException(e.getMessage());
413             }
414         }
415     }
416
417 -firstTagTest.jsp
418     <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
419     <%@ taglib prefix="easy" uri="/WEB-INF/tlds/mytags.tld" %>
420     <!DOCTYPE html>
421     <html lang="ko">
422     <head>
423     <meta charset="UTF-8">
424     <title>My First Simple Tag</title>
425     </head>
426     <body>
427         Hello^^ <br />
428         <easy:firstTag />
429     </body>
430 </html>
431
432 -mytags.tld
433     <?xml version="1.0" encoding="UTF-8"?>
434     <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
435         xmlns="http://java.sun.com/xml/ns/j2ee"
436         xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_3_0.xsd"
437         version="3.0">
438         <description>JSP 2.0 Simple Tag Demos</description>
439         <tlib-version>1.0.0.0</tlib-version>
440         <short-name>My First Taglib Example</short-name>
441
442         <tag>
443             <name>firstTag</name>
444             <tag-class>kr.co.javaexpert.libs.MyFirstTag</tag-class>
445             <body-content>empty</body-content>
446         </tag>
447     </taglib>
448
449 -MyFirstTag.java
450     package com.example.libs;
451
452     import java.io.IOException;
453     import javax.servlet.jsp.JspContext;
454     import javax.servlet.jsp.JspException;
455     import javax.servlet.jsp.tagext.JspFragment;
456     import javax.servlet.jsp.tagext.JspTag;
457     import javax.servlet.jsp.tagext.SimpleTag;
458
459     public class MyFirstTag implements SimpleTag{
460         JspContext jspContext;
461
462         public void doTag() throws JspException, IOException{
463             System.out.println("Called doTag()");
464             jspContext.getOut().print("내 첫 태그");
465         }
466

```

```

467     public void setParent(JspTag parent){
468         System.out.println("Called setParent()");
469     }
470
471     public JspTag getParent(){
472         System.out.println("Called getParent()");
473         return null;
474     }
475
476     public void setJspContext(JspContext jspContext){
477         System.out.println("Called setJspContext()");
478         this.jspContext = jspContext;
479     }
480
481     public void setJspBody(JspFragment body){
482         System.out.println("Called setJspBody()");
483     }
484 }

```

<http://localhost:8080/CustomTagDemo/firstTagTest.jsp>

-예상 출력

setJspContext

doTag

-firstTagTest.jsp에서 태그가 바디를 갖지 않기 때문에 doTag 메소드 전에 setJspBody 메소드가 호출되지 않는다.

-심플 태그가 다른 태그의 내부에 사용되지 않았기 때문에 태그 핸들러의 setParent 메소드도 호출되지 않는다.

## 6. Tag Style

1)<br /> : body 도 없고, 속성도 없는 태그

-simpleTagDemo.jsp

```

497 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
498 <%@ taglib prefix="simple" uri="/WEB-INF/tlds/welcome.tld" %>
499 <!DOCTYPE html>
500 <html lang="ko">
501 <head>
502 <meta charset="UTF-8">
503 <title>SimpleTag Demo</title>
504 </head>
505 <body>
506 <h1>JSP 2.0 Style Custom Tag</h1>
507 <simple:welcome />
508 </body>
509 </html>

```

-welcome.tld

```

512 <?xml version="1.0" encoding="UTF-8"?>
513 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
514         xmlns="http://java.sun.com/xml/ns/j2ee"
515         xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_3_0.xsd"
516         version="3.0">
517     <description>JSP 2.0 Style</description>
518     <tlib-version>1.0.0.0</tlib-version>
519     <short-name>simple</short-name>
520
521     <tag>
522         <name>welcome</name>
523         <tag-class>kr.co.javaexpert.libs.SimpleTag</tag-class>
524         <body-content>empty</body-content>
525     </tag>
526 </taglib>

```

-SimpleTag.java

```

529 package com.example.libs;
530
531 import javax.servlet.jsp.JspException;
532 import javax.servlet.jsp.JspWriter;
533 import javax.servlet.jsp.tagext.SimpleTagSupport;

```



```

534
535 public class SimpleTag extends SimpleTagSupport{
536     public void doTag() throws JspException, java.io.IOException{
537         JspWriter out = this.getJspContext().getOut();
538         try{
539             out.print("<font size='7' color='red'>");
540             out.print("Hello, Custome Tag 2.0");
541             out.println("</font>");
542         }catch(Exception ex){
543             throw new JspException(ex.getMessage());
544         }
545     }
546 }

```

2)<hr size='2' color='red' width='50%' align='left' /> :

-속성은 있지만, body 가 없는 태그

-simpleTagDemo1.jsp

```

551 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
552 <%@ taglib prefix="easy" uri="/WEB-INF/tlds/mytags.tld" %>
553 <!DOCTYPE html>
554 <html lang="ko">
555 <head>
556 <meta charset="UTF-8">
557 <title>Custom Tag Demo</title>
558 </head>
559 <body>
560 <easy:dataFormatter header="States" items="Alabama, Alaska, Georgia, Florida" /> <br />
561 <easy:dataFormatter header="Countries">
562     <jsp:attribute name="items">
563         US, UK, Canada, Korea
564     </jsp:attribute>
565 </easy:dataFormatter>
566 </body>
567 </html>

```

-mytags.tld

```

570 <?xml version="1.0" encoding="UTF-8"?>
571 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
572     xmlns="http://java.sun.com/xml/ns/j2ee"
573     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd"
574     version="2.0">
575     <tlib-version>1.0</tlib-version>
576     <short-name>easy</short-name>
577
578     <tag>
579         <name>dataFormatter</name>
580         <tag-class>kr.co.javaexpert.libs.DataFormatterTag</tag-class>
581         <body-content>empty</body-content>
582         <attribute>
583             <name>header</name>
584             <required>true</required>
585         </attribute>
586         <attribute>
587             <name>items</name>
588             <required>true</required>
589         </attribute>
590     </tag>
591     <tag>
592         <name>select</name>
593         <tag-class>kr.co.javaexpert.libs.SelectElementTag</tag-class>
594         <body-content>scriptless</body-content>
595     </tag>
596 </taglib>

```

-DataFormatterTag.java

```

package com.example.libs;

```

```

601 import java.io.IOException;
602 import java.util.StringTokenizer;
603
604 import javax.servlet.jsp.JspContext;
605 import javax.servlet.jsp.JspException;
606 import javax.servlet.jsp.JspWriter;
607 import javax.servlet.jsp.tagext.SimpleTagSupport;
608
609 public class DataFormatterTag extends SimpleTagSupport {
610     private String header;
611     private String items;
612
613     public void setHeader(String header) {
614         this.header = header;
615     }
616     public void setItems(String items) {
617         this.items = items;
618     }
619     public void doTag() throws IOException, JspException{
620         JspContext jspContext = this.getJspContext();
621         JspWriter out = jspContext.getOut();
622
623         out.println("<table style='border:1px solid green'>" +
624             "<tr><td><span style='font-weight:bold'>" +
625             header + "</span></td></tr>");
626         StringTokenizer st = new StringTokenizer(items, ",");
627         while(st.hasMoreTokens()){
628             String token = st.nextToken();
629             out.println("<tr><td>" + token + "</td></tr>");
630         }
631         out.println("</table>");
632     }
633 }

```

2)<div>Hello</div> :

-Tag Body 가 있는 태그

-simpleTagDemo2.jsp

```

638 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
639 <%@ taglib prefix="easy" uri="/WEB-INF/tlds/mytags.tld" %>
640 <!DOCTYPE html>
641 <html lang="ko">
642 <head>
643 <meta charset="UTF-8">
644 <title>Custom Tag Demo</title>
645 </head>
646 <body>
647     <easy:select>
648         <option value="{value}"> ${text}</option>
649     </easy:select>
650 </body>
651 </html>

```

-SelectElementTag.java

```

654 package com.example.libs;
655
656 import java.io.IOException;
657
658 import javax.servlet.jsp.JspContext;
659 import javax.servlet.jsp.JspException;
660 import javax.servlet.jsp.JspWriter;
661 import javax.servlet.jsp.tagext.SimpleTagSupport;
662
663 public class SelectElementTag extends SimpleTagSupport {
664     private String [] countries = {"대한민국", "브라질", "호주"};
665
666     public void doTag() throws IOException, JspException{
667         JspContext jspContext = this.getJspContext();

```

```

668     JspWriter out = jspContext.getOut();
669     out.print("<select>");
670     for(int i = 0 ; i < countries.length ; i++){
671         this.getJspContext().setAttribute("value", countries[i]);
672         this.getJspContext().setAttribute("text", countries[i]);
673         this.getJspBody().invoke(null);
674         //invoke() 에 null 을 넘기면 출력 결과는 getOut() 메소드가 반환하는
675         //JspWriter 에 전달
676     }
677     out.println("</select>");
678 }
679 }

```

## 681 7. EL 함수 사용하기

- 682 1)EL에서 함수를 호출할 수 있다.
- 683 2)정적 메소드를 가진 public 클래스를 만든다.
- 684 3)function 요소를 이용해서 태그 라이브러리 기술서에 함수를 등록한다.
- 685 4)기술서에 다음의 하위 요소를 가진다.
  - 686 -description : 태그 정보로, 생략 가능
  - 687 -display-name : XML 도구에서 출력될 짧은 이름
  - 688 -icon : XML 도구에서 사용할 아이콘 요소
  - 689 -name : 함수의 고유 이름
  - 690 -function-class : 함수를 구현한 자바 클래스의 패키지명을 포함한 클래스명
  - 691 -function-signature : 함수에 해당하는 정적 메소드의 시그니처
  - 692 -example : 함수의 사용 예로, 생략 가능
  - 693 -function-extension : XML 도구에서 사용할 함수의 추가 정보. 원하는 만큼 추가 가능

## 694 5)Syntax

```
695 ${prefix:functionName(paremeterList)}
```

## 697 6)Lab

```

698 -simpleTagDemo3.jsp
699 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
700 <%@ taglib prefix="f" uri="/WEB-INF/tlds/functiontags.tld" %>
701 <!DOCTYPE html>
702 <html lang="ko">
703 <head>
704 <meta charset="UTF-8">
705 <title>Custom Tag Demo</title>
706 </head>
707 <body>
708     ${f:reverseString("Hello World")}
709 </body>
710 </html>

```

## 712 -functiontags.tld

```

713 <?xml version="1.0" encoding="UTF-8"?>
714 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
715         xmlns="http://java.sun.com/xml/ns/j2ee"
716         xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd"
717         version="2.0">
718     <tlib-version>1.0</tlib-version>
719     <function>
720         <description>Reverses a String</description>
721         <name>reverseString</name>
722         <function-class>kr.co.javaexpert.libs.StringFunction</function-class>
723         <function-signature>java.lang.String reverseString(java.lang.String)</function-signature>
724     </function>
725 </taglib>

```

## 727 -StringFunction.java

```

728 package com.example.libs;
729
730 public class StringFunction {
731     public static String reverseString(String s){
732         return new StringBuffer(s).reverse().toString();
733     }
734 }

```

735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801

8. Custom Tag 배포

- 1) Custom Tag Handler 와 Custom Tag Descriptor 는 JSTL 과 같이 jar 파일로 묶어 배포가능
- 2) 이 경우 jar 안에 모든 태그 핸들러와 tld 파일을 넣어야 한다.
- 3) 기술서의 uri 요소에서 URI 를 지정할 때 절대 경로를 넣어야 한다.
- 4) mytags.jar

```
|
-----function
|
-----StringFunction.class
|
-----META-INF
|
-----functiontags.tld
```

5) functiontags.tld

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd"
  version="2.0">
  <description>Funtion tag examples</description>
  <tlib-version>1.0</tlib-version>

  <url>http://javaexpert.co.kr/taglib/function</uri> <---아주 중요

  <function>
    <description>Reverses a String</description>
    <name>reverseString</name>
    <function-class>kr.co.javaexpert.libs.StringFunction</function-class>
    <function-signature>java.lang.String reverseString(java.lang.String)</function-signature>
  </function>

</taglib>
```

6)jar 파일을 WEB-INF/lib 디렉토리로 복사

7) 사용할 jsp 페이지의 taglib 속성은 아래와 같이 지정해야 함

```
<%@ taglib uri="http://javaexpert.co.kr/taglib/function" prefix="f" %>
- 즉 tld 파일속의 uri 와 같아야 함.
```

9. JSP 2.0 또는 그 이상의 버전에서 Tag 파일을 이용해서 구현한 커스텀 태그

1)개요

- JSP page
- Tag file
  - a. tld 파일 대체
  - b. Custom tag 당 하나의 파일이어야 한다.

2)작성방법

- Custom tag 정의한다.
- 확장자를 .tag를 사용하는 파일생성
- <%@ tag %>를 사용하는 directive
- 나머지는 jsp 페이지의 문법과 비슷

3)Lab

```
-TagDemo.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset=UTF-8">
<title>커스텀 태그 연습</title>
</head>
<body>
<h1>JSP 2.0 이상 환경에서 Tag 파일을 이용해서 구현한 커스텀 태그</h1>
<aaa:bbb />
```

```
802     </body>
803     </html>
804
805 -bbb.tag
806     <%@ tag body-content="empty" pageEncoding="utf-8" %>
807     <font size='7' color='red'>안녕하세요. Custom Tag 2.0</font>
808
```

### 3) tag 파일에서 사용할 수 있는 내장객체들

- a. request : javax.servlet.http.HttpServletRequest
- b. response : javax.servlet.http.HttpServletResponse
- c. out : javax.servlet.jsp.JspWriter
- d. session : javax.servlet.http.HttpSession
- e. application : javax.servlet.ServletContext
- f. config : javax.servlet.ServletConfig
- g. jspContext : javax.servlet.jsp.JspContext

### 4) tag 파일의 directive 들

- a. taglib : jsp 페이지의 taglib 와 동일한 디렉티브
- b. include
- c. tag : jsp 페이지의 page 디렉티브와 유사
- d. attribute : 태그 파일에서 커스텀 태그의 속성을 선언(명시)
- e. variable : 표현 언어(EL) 변수를 선언(명시)

### 5) tag directive 의 속성들

- display-name : tag file 의 확장자를 제외한 파일이름
- body-content : custom tag 의 body에 대한 정보를 제공,  
empty | tagdependent | scriptless(기본값) 중 하나 선택
- dynamic-attributes : custom tag 의 동적 속성 지정, map 방식으로 지정해야 함
- small-icon
- large-icon
- description
- language
- import
- pageEncoding

### 6)attribute 속성

- name : 태그 파일에서 사용할 속성 이름, 고유해야 함
- required : 필수 속성 여부, 기본값은 false
- fragment : 속성이 태그 핸들러에서 처리하는 프래그먼트 속성인지, 태그 핸들러에 전달하기 전에 컨테이너가 미리 평가하는 일반 속성인지 여부. 기본값은 false. 만일 태그 핸들러에서 속성을 처리하는 경우에는 값을 true 로 지정해야 함.
- rtexprvalue : 속성 값이 스크립틀릿 표현식으로 실행 시간에 동적으로 계산되는지 여부, 기본값은 true
- type : 속성 값의 타입. 기본값은 java.lang.String
- description : 속성에 대한 설명

### 7)Lab

```
846 -TagDemo1.jsp
847     <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
848     <%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
849     <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
850     <html>
851     <head>
852     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
853     <title>Insert title here</title>
854     </head>
855     <body>
856     <b><aaa:ccc>여기는 태그바디...</aaa:ccc></b>
857     </body>
858     </html>
859
```

- body 가 있는 경우 body-content의 값은 tagdependent나 scriptless로 지정해야
- 속성값이 tagdependent 의 경우는 커스텀 태그의 내용을 처리하지 않고 그대로 사용  
scriptless일 경우 EL 요소나 액션 태그의 처리 결과를 사용
- <jsp:doBody />를 사용해서 body 내용을 처리해야

```
866 -ccc.tag
867     <%@ tag body-content="scriptless" pageEncoding="utf-8" %>
```

```

868 <%@ tag import="java.util.Date" %>
869 <%
870     Date now = new Date();
871 %>
872 Now is <%=now %>
873 <jsp:doBody />
874
875 -TagDemo2.jsp
876 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
877 <%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
878 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
879 <html>
880 <head>
881 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
882 <title>Insert title here</title>
883 </head>
884 <body>
885     <aaa:ddd title="os" bgcolor="#ededed" color="#ffffff">
886         Windows<br />
887         Linux<br />
888         Unix
889     </aaa:ddd>
890 </body>
891 </html>

```

```

893 -ddd.tag
894 <%@ tag body-content="scriptless" pageEncoding="utf-8" %>
895 <%@ attribute name="color" %>
896 <%@ attribute name="bgcolor" %>
897 <%@ attribute name="title" %>
898 <table border="1" bgcolor="${color}">
899     <tr>
900         <td>${title}</td>
901     </tr>
902     <tr>
903         <td bgcolor="${bgcolor}">
904             <jsp:doBody />
905         </td>
906     </tr>
907 </table>

```

## 7) variable 지시어

-JSP 페이지에 태그 파일의 값을 노출하고자 하는 경우 사용

-속성

- a. name-given : 호출 JSP 페이지의 스크립팅 언어나 EL 표현식에 제공할 변수명. name-from-attribute 와 함께 사용할 수 없음. 태그 파일 내의 속성 이름과 같은 이름은 값으로 쓸 수 없음.
- b. name-from-attribute : 속성의 값으로 지정한 태그 파일 속성의 값이 변수명이 된다.
- c. alias : 변수의 값을 가진 지역 유효범위 속성의 이름.
- d. variable-class : 변수의 타입. 기본타입은 java.lang.String.
- e. declare : 변수가 호출 페이지에 선언됐는지, 태그 파일에 선언됐는지 여부. 기본 값은 true로, 태그 파일에 선언됐음을 의미함.
- f. scope : 유효범위. 값은 AT\_BEGIN, AT\_END, NESTED 중 하나여야 함. 기본값은 NESTED.
- g. description : 설명.

-vardemo.jsp

```

921 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
922 <%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
923 오늘 날짜 : <tags:varDemo>
924 긴 포맷 : ${longDate}<br />
925 짧은 포맷 : ${shortDate}
926 </tags:varDemo>

```

-varDemo.tag

```

929 <%@ tag import="java.util.Date" import="java.text.DateFormat" pageEncoding="utf-8" %>
930 <%@ variable name-given="longDate" %>
931 <%@ variable name-given="shortDate" %>
932 <%

```

```

933     Date now = new Date(System.currentTimeMillis());
934     DateFormat longFormat = DateFormat.getDateInstance(DateFormat.LONG);
935     DateFormat shortFormat = DateFormat.getDateInstance(DateFormat.SHORT);
936     jspContext.setAttribute("longDate", longFormat.format(now));
937     jspContext.setAttribute("shortDate", shortFormat.format(now));
938 %>
939 <jsp:doBody />
940

```

#### 8) doBody action

- doBody 는 tag 파일안에서만 사용가능
- 태그의 바디를 호출할 때 사용
- 속성을 가질 수 있으며, 태그 호출 결과를 변수에 저장도 가능하다. 만일 저장할 경우 이 속성을 이용한다.
- 속성을 사용하지 않을 경우 doBody 액션은 결과를 JspWriter 로 출력한다.
- 속성
  - a. var : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.lang.String 타입으로 저장. var와 varReader 함께 사용 못함.
  - b. varReader : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.io.Reader 타입으로 저장. var와 varReader 함께 사용 못함.
  - c. scope : 결과 값의 유효범위

#### 9)invoke action

- 표준 액션 invoke 는 doBody 와 비슷한 동작을 하며, 태그 파일에서 프래그먼트 속성을 호출할 때 사용
- fragment 속성의 값이 true이면 해당 속성은 프래그먼트 속성이 되며, 태그 파일에서 여러번 호출할 수 있다.
- invoke 도 속성을 가지며, 속성은 아래와 같다.
  - a. fragment : 프래그먼트를 식별하기 위한 이름
  - b. var : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.lang.String 타입으로 저장. var와 varReader 함께 사용 못함.
  - c. varReader : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.io.Reader 타입으로 저장. var와 varReader 함께 사용 못함.
  - d. scope : 결과 값의 유효범위

#### -invokeTest.jsp

```

961 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
962 <%@ taglib prefix="easy" tagdir="/WEB-INF/tags" %>
963 <!DOCTYPE html>
964 <html lang="ko">
965 <head>
966 <meta charset="UTF-8">
967 <title>Invoke Demo</title>
968 </head>
969 <body>
970 <easy:invokeDemo>
971 <jsp:attribute name="productDetails">
972 <table width="220" border="1">
973 <tr>
974 <td><b>상품명</b></td>
975 <td>${productName}</td>
976 </tr>
977 <tr>
978 <td><b>설명</b></td>
979 <td>${description}</td>
980 </tr>
981 <tr>
982 <td><b>가격</b></td>
983 <td>${price}</td>
984 </tr>
985 </table>
986 </jsp:attribute>
987 </easy:invokeDemo>
988 </body>
989 </html>
990

```

#### -invokeDemo.tag

```

992 <%@ tag pageEncoding="utf-8" %>
993 <%@ attribute name="productDetails" fragment="true" %>
994 <%@ variable name-given="productName" %>
995 <%@ variable name-given="description" %>

```

```
996 <%@ variable name-given="price" %>
997 <%
998     jspContext.setAttribute("productName", "오성전자 DVD 플레이어");
999     jspContext.setAttribute("description", "동축 디지털 오디오 잭을 통한 돌비 디지털 출력, " +
1000         "풀 HD 화질 재생");
1001     jspContext.setAttribute("price", "65");
1002 %>
1003 <jsp:include fragment="productDetails" />
```