

```

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

```

```

67 public class HelloWorld extends TagSupport{
68     @Override
69     public int doStartTag() throws JspTagException{
70         JspWriter out = pageContext.getOut();
71         try {
72             out.print("<font size='7' color='red'>");
73             out.println("Hello, World</font>");
74         } catch (Exception e) {
75             throw new JspTagException(e.getMessage());
76         }
77         return this.SKIP_BODY;
78     }
79     @Override
80     public int doEndTag() throws JspTagException{
81         return this.EVAL_PAGE;
82     }
83 }
84
85 2) javax.servlet.jsp.tagext.Tag interface or javax.servlet.jsp.tagext.TagSupport class
86 - int EVAL_BODY_INCLUDE
87 - int SKIP_BODY
88 - int EVAL_PAGE
89 - int SKIP_PAGE
90 - public int doStartTag() throws JspException
91     a. 시작 태그를 만날 때 실행
92 - public int doEndTag() throws JspException
93     a. 끝 태그를 만날 때 실행
94 - public void setPageContext(PageContext pageContext)
95     a. 전달 받은 pageContext 객체를 저장할 때 사용
96
97 3)Tag Style
98     a. <br /> : body 도 없고, 속성도 없는 태그
99
100 *****date.jsp*****
101 <?xml version="1.0" encoding="UTF-8" ?>
102 <%@ page language="java" contentType="text/html; charset=UTF-8"
103     pageEncoding="UTF-8"%>
104 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mydate.tld" %>
105 <body>
106     Current time is <mylib:date />.
107 </body>
108 *****mydate.tld*****
109 <?xml version="1.0" encoding="UTF-8" ?>
110
111 <taglib xmlns="http://java.sun.com/xml/ns/j2ee"
112     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
113     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
114     http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
115     version="2.0">
116     <tlib-version>1.0</tlib-version>
117     <short-name>mylib</short-name>
118     <tag>
119         <name>date</name>
120         <tag-class>kr.co.javaexpert.libs.MyDate</tag-class>
121         <body-content>empty</body-content>
122     </tag>
123 </taglib>
124 *****MyDate.java*****
125 package kr.co.javaexpert.libs;
126
127 import java.util.Date;
128
129 import javax.servlet.ServletException;
130 import javax.servlet.jsp.JspTagException;
131 import javax.servlet.jsp.JspWriter;
132 import javax.servlet.jsp.tagext.TagSupport;
133

```

```

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

```

152  
153 b. <hr size='2' color='red' width='50%' align='left' /> :  
154 속성은 있지만, body 가 없는 태그  
155

```

156 *****date2.jsp*****
157 <%@ page language="java" contentType="text/html; charset=UTF-8"
158     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*****
186 <?xml version="1.0" encoding="UTF-8" ?>
187
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
191         http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
192     version="2.0">
193     <tlib-version>2.0</tlib-version>
194     <short-name>mylib</short-name>
195     <tag>
196         <name>date</name>
197         <tag-class>kr.co.javaexpert.libs.MyDate</tag-class>
198         <body-content>empty</body-content>
199     </tag>
200 </taglib>

```

```

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 *****MyDate2.java*****
227 package kr.co.javaexpert.libs;
228
229 import java.text.DateFormat;
230 import java.util.Date;
231 import java.util.Locale;
232
233 import javax.servlet.jsp.JspException;
234 import javax.servlet.jsp.JspWriter;
235 import javax.servlet.jsp.tagext.TagSupport;
236
237 public class MyDate2 extends TagSupport {
238     private String type;
239     public void setType(String type){
240         this.type = type;
241     }
242     public int doStartTag() throws JspException{
243         JspWriter out = pageContext.getOut();
244         DateFormat df = null;
245         try{
246             Date current = new Date();
247             Locale locale = null;
248             if(type == null) type="0";
249             switch(type){
250                 case "0" : locale = Locale.KOREA; break;
251                 case "1" : locale = Locale.JAPAN; break;
252                 case "2" : locale = Locale.US; break;
253                 case "3" : locale = Locale.UK; break;
254                 case "4" : locale = Locale.CHINA; break;
255                 case "5" : locale = Locale.FRENCH; break;
256             }
257             df = DateFormat.getDateTimeInstance(
258                 DateFormat.FULL, DateFormat.FULL, locale);
259             out.println(df.format(current));
260         }catch(Exception ex){
261             throw new JspException(ex.getMessage());
262         }
263         return SKIP_BODY;
264     }
265     public int doEndTag() throws JspException{
266         type = "0";

```

```

267         return EVAL_PAGE;
268     }
269 }
270
271 *****header.jsp*****
272 <?xml version="1.0" encoding="UTF-8" ?>
273 <%@ page language="java" contentType="text/html; charset=UTF-8"
274     pageEncoding="UTF-8"%>
275 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mylib.tld" %>
276 <body>
277     <h2>Request Headers</h2>
278     1. user-agent : <mylib:header name="user-agent" /><br />
279     2. accept : <mylib:header name="accept" /><br />
280     3. cookie : <mylib:header name="cookie" /><br />
281 </body>
282 *****MyHeader.java*****
283 package kr.co.javaexpert.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 c. <font size='5' color='red' face='Arial'>Hello</font> :
315     속성도 있고, body 도 있는 태그
316
317 *****content.jsp*****
318 <?xml version="1.0" encoding="UTF-8" ?>
319 <%@ page language="java" contentType="text/html; charset=UTF-8"
320     pageEncoding="UTF-8"%>
321 <%@ taglib prefix="mylib" uri="/WEB-INF/tlds/mylib.tld" %>
322 <body>
323     <h3>Tag Body 연습</h3>
324     <mylib:content>
325         구성모
326     </mylib:content>
327 </body>
328 *****MyContent.java*****
329 package kr.co.javaexpert.libs;
330
331 import javax.servlet.jsp.JspTagException;
332 import javax.servlet.jsp.JspWriter;
333 import javax.servlet.jsp.tagext.TagSupport;

```

```

334
335 public class MyContent extends TagSupport {
336     @Override
337     public int doStartTag() throws JspTagException{
338         try {
339             pageContext.getOut().print("반갑습니다.");
340         } catch (Exception e) {
341             throw new JspTagException(e.getMessage());
342         }finally{
343             return this.EVAL_BODY_INCLUDE;
344         }
345     }
346     @Override
347     public int doEndTag() throws JspTagException{
348         try{
349             pageContext.getOut().println("님! ^-^;");
350         }catch(Exception ex){
351             throw new JspTagException(ex.getMessage());
352         }finally{
353             return this.EVAL_PAGE;
354         }
355     }
356 }
357

```

4) web.xml 코드 (<--JSP 2.x 이상에서는 필요없음)

```

358 <taglib>
359     <taglib-uri>/WEB-INF/tlds/ccc.tld</taglib-uri>
360     <taglib-location></taglib-location>
361 </taglib>
362

```

4. JSP 2.0 또는 그 이상의 버전에서 구현한 커스텀 태그

- JSP 1.1 과 JSP 1.2의 커스텀 태그와 태그 핸들러가 작성하기 너무 어려웠다.
- 그래서 JSP 2.0 에서 javax.servlet.jsp.tagext의 SimpleTag interface 추가
- SimpleTag 를 구현한 태그 핸들러는 Simple Tag Handler 라 부른다.
- 기존 Tag, IterationTag, BodyTag 를 구현한 태그 핸들러는 클래식 태그 핸들러라고 부른다.
- 작성하기 쉽고, 생명주기도 단순하다.
- 태그 실행과 관련된 메소드는 doTag 하나뿐
- 이 메소드는 태그를 실행할 때 한번 호출
- 심플 태그 핸들러의 생명주기
  - a. JSP Container 는 인자 없는 생성자를 호출해서 인스턴스를 생성, 따라서 반드시 인자 없는 생성자를 가져야 함.
  - b. 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.
 

```

375     public void setJspContext(JspContext jspContext)
376

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

```

377     public void setParent(JspTag parent)
378

```
  - d. Jsp Container 는 태그의 각 속성별 setter 메소드를 호출한다.
  - e. 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.
  - f. Jsp Container 는 doTag 메소드를 호출한다. doTag 메소드가 리턴할 때 모든 변수는 동기화된 상태이다.

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

2) Java class 수정

```

386 ///////////////////////////////////////////////////
387 public class HelloWorld extends TagSupport{
388     @Override
389     public int doStartTag() throws JspTagException{
390         JspWriter out = pageContext.getOut();
391         try {
392             out.print("<font size='7' color='red'>");
393             out.println("Hello, World</font>");
394         } catch (Exception e) {
395             throw new JspTagException(e.getMessage());
396         }
397         return this.SKIP_BODY;

```

```

398     }
399     @Override
400     public int doEndTag() throws JspTagException{
401         return this.EVAL_PAGE;
402     }
403 }
404 //////////////////////////////////////
405 public class HelloWorld extends SimpleTagSupport{
406     @Override
407     public void doTag() throws JspException, java.io.IOException{
408         JspWriter out = pageContext.getOut();
409         try {
410             out.print("<font size='7' color='red'>");
411             out.println("Hello, World</font>");
412         } catch (Exception e) {
413             throw new JspTagException(e.getMessage());
414         }
415     }
416 }
417
418 *****firstTagTest.jsp*****
419 <%@ page language="java" contentType="text/html; charset=UTF-8"
420     pageEncoding="UTF-8"%>
421 <%@ taglib prefix="easy" uri="/WEB-INF/tlds/mytags.tld" %>
422 <!DOCTYPE html>
423 <html lang="ko">
424 <head>
425 <meta charset="UTF-8">
426 <title>My First Simple Tag</title>
427 </head>
428 <body>
429 Hello^^<br />
430 <easy:firstTag />
431 </body>
432 </html>
433 *****mytags.tld*****
434 <?xml version="1.0" encoding="UTF-8"?>
435
436 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
437     xmlns="http://java.sun.com/xml/ns/j2ee"
438     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_3_0.xsd"
439     version="3.0">
440     <description>JSP 2.0 Simple Tag Demos</description>
441     <tlib-version>1.0.0.0</tlib-version>
442     <short-name>My First Taglib Example</short-name>
443
444     <tag>
445         <name>firstTag</name>
446         <tag-class>kr.co.javaexpert.libs.MyFirstTag</tag-class>
447         <body-content>empty</body-content>
448     </tag>
449 </taglib>
450 *****MyFirstTag.java*****
451 package kr.co.javaexpert.libs;
452
453 import java.io.IOException;
454 import javax.servlet.jsp.JspContext;
455 import javax.servlet.jsp.JspException;
456 import javax.servlet.jsp.tagext.JspFragment;
457 import javax.servlet.jsp.tagext.JspTag;
458 import javax.servlet.jsp.tagext.SimpleTag;
459
460 public class MyFirstTag implements SimpleTag{
461     JspContext jspContext;
462
463     public void doTag() throws JspException, IOException{
464         System.out.println("Called doTag()");

```

```

465     jspContext.getOut().print("내 첫 태그");
466 }
467
468 public void setParent(JspTag parent){
469     System.out.println("Called setParent()");
470 }
471
472 public JspTag getParent(){
473     System.out.println("Called getParent()");
474     return null;
475 }
476
477 public void setJspContext(JspContext jspContext){
478     System.out.println("Called setJspContext()");
479     this.jspContext = jspContext;
480 }
481
482 public void setJspBody(JspFragment body){
483     System.out.println("Called setJspBody()");
484 }
485 }
486 http://localhost:8080/CustomTagDemo/firstTagTest.jsp
487 -예상 출력
488     setJspContext
489     doTag
490 -firstTagTest.jsp에서 태그가 바디를 갖지 않기 때문에 doTag 메소드 전에 setJspBody 메소드가 호출되지 않는다.
491 -심플 태그가 다른 태그의 내부에 사용되지 않았기 때문에 태그 핸들러의 setParent 메소드도 호출되지 않는다.
492
493 3)Tag Style
494 a. <br /> : body 도 없고, 속성도 없는 태그
495 *****simpleTagDemo.jsp*****
496 <%@ page language="java" contentType="text/html; charset=UTF-8"
497     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>
510 *****welcome.tld*****
511 <?xml version="1.0" encoding="UTF-8"?>
512
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>
527 *****SimpleTag.java*****
528 package kr.co.javaexpert.libs;
529
530 import javax.servlet.jsp.JspException;
531 import javax.servlet.jsp.JspWriter;

```



```

532 import javax.servlet.jsp.tagext.SimpleTagSupport;
533
534 public class SimpleTag extends SimpleTagSupport{
535     public void doTag() throws JspException, java.io.IOException{
536         JspWriter out = this.getJspContext().getOut();
537         try{
538             out.print("<font size='7' color='red'>");
539             out.print("Hello, Custome Tag 2.0");
540             out.println("</font>");
541         }catch(Exception ex){
542             throw new JspException(ex.getMessage());
543         }
544     }
545 }
546
547 b. <hr size='2' color='red' width='50%' align='left' /> :
548     속성은 있지만, body 가 없는 태그
549 *****simpleTagDemo1.jsp*****
550 <%@ page language="java" contentType="text/html; charset=UTF-8"
551     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>
568 *****mytags.tld*****
569 <?xml version="1.0" encoding="UTF-8"?>
570
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>
597 *****DataFormatterTag.java*****
598 package kr.co.javaexpert.libs;

```

```

599
600 import java.io.IOException;
601 import java.util.StringTokenizer;
602
603 import javax.servlet.jsp.JspContext;
604 import javax.servlet.jsp.JspException;
605 import javax.servlet.jsp.JspWriter;
606 import javax.servlet.jsp.tagext.SimpleTagSupport;
607
608 public class DataFormatterTag extends SimpleTagSupport {
609     private String header;
610     private String items;
611
612     public void setHeader(String header) {
613         this.header = header;
614     }
615     public void setItems(String items) {
616         this.items = items;
617     }
618     public void doTag() throws IOException, JspException{
619         JspContext jspContext = this.getJspContext();
620         JspWriter out = jspContext.getOut();
621
622         out.println("<table style='border:1px solid green'>" +
623             "<tr><td><span style='font-weight:bold'>" +
624             header + "</span></td></tr>");
625         StringTokenizer st = new StringTokenizer(items, ",");
626         while(st.hasMoreTokens()){
627             String token = st.nextToken();
628             out.println("<tr><td>" + token + "</td></tr>");
629         }
630         out.println("</table>");
631     }
632 }
633
634 c. <div>Hello</div> :
635     Tag Body 가 있는 태그
636 *****simpleTagDemo2.jsp*****
637 <%@ page language="java" contentType="text/html; charset=UTF-8"
638     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>
652 *****SelectElementTag.java*****
653 package kr.co.javaexpert.libs;
654
655 import java.io.IOException;
656
657 import javax.servlet.jsp.JspContext;
658 import javax.servlet.jsp.JspException;
659 import javax.servlet.jsp.JspWriter;
660 import javax.servlet.jsp.tagext.SimpleTagSupport;
661
662 public class SelectElementTag extends SimpleTagSupport {
663     private String [] countries = {"대한민국", "브라질", "호주"};
664
665     public void doTag() throws IOException, JspException{

```

```

666     JspContext jspContext = this.getJspContext();
667     JspWriter out = jspContext.getOut();
668     out.print("<select>");
669     for(int i = 0 ; i < countries.length ; i++){
670         this.getJspContext().setAttribute("value", countries[i]);
671         this.getJspContext().setAttribute("text", countries[i]);
672         this.getJspBody().invoke(null);
673         //invoke() 에 null 을 넘기면 출력 결과는 getOut() 메소드가 반환하는
674         //JspWriter 에 전달
675     }
676     out.println("</select>");
677 }
678 }
679 4)EL 함수 사용하기
680 -EL에서 함수를 호출할 수 있다.
681 -정적 메소드를 가진 public 클래스를 만든다.
682 -function 요소를 이용해서 태그 라이브러리 기술서에 함수를 등록한다.
683 -기술서에 다음의 하위 요소를 가진다.
684     --description : 태그 정보로, 생략 가능
685     --display-name : XML 도구에서 출력될 짧은 이름
686     --icon : XML 도구에서 사용할 아이콘 요소
687     --name : 함수의 고유 이름
688     --function-class : 함수를 구현한 자바 클래스의 패키지명을 포함한 클래스명
689     --function-signature : 함수에 해당하는 정적 메소드의 시그니처
690     --example : 함수의 사용 예로, 생략 가능
691     --function-extension : XML 도구에서 사용할 함수의 추가 정보. 원하는 만큼 추가 가능
692 -Syntax
693     ${prefix:functionName(paremeterList)}
694 *****simpleTagDemo3.jsp*****
695 <%@ page language="java" contentType="text/html; charset=UTF-8"
696     pageEncoding="UTF-8"%>
697 <%@ taglib prefix="f" uri="/WEB-INF/tlds/functiontags.tld" %>
698 <!DOCTYPE html>
699 <html lang="ko">
700 <head>
701 <meta charset="UTF-8">
702 <title>Custom Tag Demo</title>
703 </head>
704 <body>
705     ${f:reverseString("Hello World")}
706 </body>
707 </html>
708 *****functiontags.tld*****
709 <?xml version="1.0" encoding="UTF-8"?>
710
711 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
712     xmlns="http://java.sun.com/xml/ns/j2ee"
713     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd"
714     version="2.0">
715     <tlib-version>1.0</tlib-version>
716     <function>
717         <description>Reverses a String</description>
718         <name>reverseString</name>
719         <function-class>kr.co.javaexpert.libs.StringFunction</function-class>
720         <function-signature>java.lang.String reverseString(java.lang.String)</function-signature>
721     </function>
722 </taglib>
723
724 *****StringFunction.java*****
725 package kr.co.javaexpert.libs;
726
727 public class StringFunction {
728     public static String reverseString(String s){
729         return new StringBuffer(s).reverse().toString();
730     }
731 }
732

```

```

733 5. Custom Tag 배포
734 1) Custom Tag Handler 와 Custom Tag Descriptor 는 JSTL 과 같이 jar 파일로 묶어 배포가능
735 2) 이 경우 jar 안에 모든 태그 핸들러와 tld 파일을 넣어야 한다.
736 3) 기술서의 uri 요소에서 URI 를 지정할 때 절대 경로를 넣어야 한다.
737 4) mytags.jar
738 |
739 |-----function
740 |
741 |-----StringFunction.class
742 |
743 |-----META-INF
744 |
745 |-----functiontags.tld
746
747 5) functiontags.tld
748 <?xml version="1.0" encoding="UTF-8"?>
749
750 <taglib xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
751       xmlns="http://java.sun.com/xml/ns/j2ee"
752       xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd"
753       version="2.0">
754   <description>Funtion tag examples</description>
755   <tlib-version>1.0</tlib-version>
756
757   <url>http://javaexpert.co.kr/taglib/function</url> <---아주 중요
758
759   <function>
760     <description>Reverses a String</description>
761     <name>reverseString</name>
762     <function-class>kr.co.javaexpert.libs.StringFunction</function-class>
763     <function-signature>java.lang.String reverseString(java.lang.String)</function-signature>
764   </function>
765
766 </taglib>
767
768 6)jar 파일을 WEB-INF/lib 디렉토리로 복사
769 7) 사용할 jsp 페이지의 taglib 속성은 아래와 같이 지정해야 함
770 <%@ taglib uri="http://javaexpert.co.kr/taglib/function" prefix="f" %>
771 - 즉 tld 파일속의 uri 와 같아야 함.
772
773
774 6. JSP 2.0 또는 그 이상의 버전에서 Tag 파일을 이용해서 구현한 커스텀 태그
775 1)개요
776 - JSP page
777 - Tag file
778   a. tld 파일 대체
779   b. Custom tag 당 하나의 파일이어야 한다.
780 2)작성방법
781 - Custom tag 정의한다.
782 - 확장자를 .tag를 사용하는 파일생성
783 - <%@ tag %>를 사용하는 directive
784 - 나머지는 jsp 페이지의 문법과 비슷
785
786 *****TagDemo.jsp*****
787 <%@ page language="java" contentType="text/html; charset=UTF-8"
788   pageEncoding="UTF-8"%>
789 <%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
790 <!DOCTYPE html>
791 <html lang="ko">
792 <head>
793 <meta charset=UTF-8">
794 <title>커스텀 태그 연습</title>
795 </head>
796 <body>
797 <h1>JSP 2.0 이상 환경에서 Tag 파일을 이용해서 구현한 커스텀 태그</h1>
798 <aaa:bbb />
799 </body>

```

```

800 </html>
801 *****bbb.tag*****
802 <%@ tag body-content="empty" pageEncoding="utf-8" %>
803 <font size='7' color='red'>안녕하세요. Custom Tag 2.0</font>
804
805 3) tag 파일에서 사용할 수 있는 내장객체들
806 a. request : javax.servlet.http.HttpServletRequest
807 b. response : javax.servlet.http.HttpServletResponse
808 c. out : javax.servlet.jsp.JspWriter
809 d. session : javax.servlet.http.HttpSession
810 e. application : javax.servlet.ServletContext
811 f. config : javax.servlet.ServletConfig
812 g. jspContext : javax.servlet.jsp.JspContext
813
814 4) tag 파일의 directive 들
815 a. taglib : jsp 페이지의 taglib 와 동일한 디렉티브
816 b. include
817 c. tag : jsp 페이지의 page 디렉티브와 유사
818 d. attribute : 태그 파일에서 커스텀 태그의 속성을 선언(명시)
819 e. variable : 표현 언어(EL) 변수를 선언(명시)
820
821 5) tag directive 의 속성들
822 - display-name : tag file 의 확장자를 제외한 파일이름
823 - body-content : custom tag 의 body에 대한 정보를 제공,
824     empty | tagdependent | scriptless(기본값) 중 하나 선택
825 - dynamic-attributes : custom tag 의 동적 속성 지정, map 방식으로 지정해야 함
826 - small-icon
827 - large-icon
828 - description
829 - language
830 - import
831 - pageEncoding
832
833 6)attribute 속성
834 - name : 태그 파일에서 사용할 속성 이름, 고유해야 함
835 - required : 필수 속성 여부, 기본값은 false
836 - 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.
837 - rtexprvalue : 속성 값이 스크립틀릿 표현식으로 실행 시간에 동적으로 계산되는지 여부, 기본값은 true
838 - type : 속성 값의 타입. 기본값은 java.lang.String
839 - description : 속성에 대한 설명
840
841 *****TagDemo1.jsp*****
842 <%@ page language="java" contentType="text/html; charset=UTF-8"
843     pageEncoding="UTF-8"%>
844 <%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
845 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
846 <html>
847 <head>
848 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
849 <title>Insert title here</title>
850 </head>
851 <body>
852 <b><aaa:ccc>여기는 태그바디...</aaa:ccc></b>
853 </body>
854 </html>
855 - body 가 있는 경우 body-content의 값은 tagdependent나 scriptless로 지정해야
856 - 속성값이 tagdependent 의 경우는 커스텀 태그의 내용을 처리하지 않고 그대로 사용
857     scriptless일경우 EL 요소나 액션 태그의 처리 결과를 사용
858 - <jsp:doBody />를 사용해서 body 내용을 처리해야
859 *****ccc.tag*****
860 <%@ tag body-content="scriptless" pageEncoding="utf-8" %>
861 <%@ tag import="java.util.Date" %>
862 <%
863     Date now = new Date();
864 %>

```

```

865 Now is <%=now %>
866 <jsp:doBody />
867
868 *****TagDemo2.jsp*****
869 <%@ page language="java" contentType="text/html; charset=UTF-8"
870     pageEncoding="UTF-8"%>
871 <%@ taglib prefix="aaa" tagdir="/WEB-INF/tags" %>
872 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
873     "http://www.w3.org/TR/html4/loose.dtd">
874 <html>
875 <head>
876 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
877 <title>Insert title here</title>
878 </head>
879 <body>
880     <aaa:ddd title="os" bgcolor="#ededed" color="#ffffff">
881         Windows<br />
882         Linux<br />
883         Unix
884     </aaa:ddd>
885 </body>
886 </html>
887 *****ddd.tag*****
888 <%@ tag body-content="scriptless" pageEncoding="utf-8" %>
889 <%@ attribute name="color" %>
890 <%@ attribute name="bgcolor" %>
891 <%@ attribute name="title" %>
892 <table border="1" bgcolor="${color}">
893     <tr>
894         <td>${title}</td>
895     </tr>
896     <tr>
897         <td bgcolor="${bgcolor}">
898             <jsp:doBody />
899         </td>
900     </tr>
901 </table>
902
903 7) variable 지시어
904 - JSP 페이지에 태그 파일의 값을 노출하고자 하는 경우 사용
905 - 속성
906     a. name-given : 호출 JSP 페이지의 스크립팅 언어나 EL 표현식에 제공할 변수명. name-from-attribute 와 함께
907     사용할 수 없음. 태그 파일 내의 속성 이름과 같은 이름은 값으로 쓸 수 없음.
908     b. name-from-attribute : 속성의 값으로 지정한 태그 파일 속성의 값이 변수명이 된다.
909     c. alias : 변수의 값을 가진 지역 유효범위 속성의 이름.
910     d. variable-class : 변수의 타입. 기본타입은 java.lang.String.
911     e. declare : 변수가 호출 페이지에 선언됐는지, 태그 파일에 선언됐는지 여부. 기본 값은 true로, 태그 파일에 선언됐음을
912     의미함.
913     f. scope : 유효범위. 값은 AT_BEGIN, AT_END, NESTED 중 하나여야 함. 기본값은 NESTED.
914     g. description : 설명.
915
916 *****vardemo.jsp*****
917 <%@ page language="java" contentType="text/html; charset=UTF-8"
918     pageEncoding="UTF-8"%>
919 <%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
920 오늘 날짜 : <tags:varDemo>
921 긴 포맷 : ${longDate}<br />
922 짧은 포맷 : ${shortDate}
923 </tags:varDemo>
924 *****varDemo.tag*****
925 <%@ tag import="java.util.Date" import="java.text.DateFormat"
926     pageEncoding="utf-8" %>
927 <%@ variable name-given="longDate" %>
928 <%@ variable name-given="shortDate" %>
929 <%
930     Date now = new Date(System.currentTimeMillis());
931     DateFormat longFormat = DateFormat.getDateInstance(DateFormat.LONG);

```

```

929     DateFormat shortFormat = DateFormat.getDateInstance(DateFormat.SHORT);
930     jspContext.setAttribute("longDate", longFormat.format(now));
931     jspContext.setAttribute("shortDate", shortFormat.format(now));
932 %>
933 <jsp:doBody />
934
935 8) doBody action
936 - doBody 는 tag 파일안에서만 사용가능
937 - 태그의 바디를 호출할 때 사용
938 - 속성을 가질 수 있으며, 태그 호출 결과를 변수에 저장도 가능하다. 만일 저장할 경우 이 속성을 이용한다.
939 - 속성을 사용하지 않을 경우 doBody 액션은 결과를 JspWriter 로 출력한다.
940 - 속성
941     a. var : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.lang.String 타입으로 저장. var와
        varReader 함께 사용 못함.
942     b. varReader : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.io.Reader 타입으로 저장. var와
        varReader 함께 사용 못함.
943     c. scope : 결과 값의 유효범위
944
945 9)invoke action
946 - 표준 액션 invoke 는 doBody 와 비슷한 동작을 하며, 태그 파일에서 프래그먼트 속성을 호출할 때 사용
947 - fragment 속성의 값이 true이면 해당 속성은 프래그먼트 속성이 되며, 태그 파일에서 여러번 호출할 수 있다.
948 - invoke 도 속성을 가지며, 속성은 아래와 같다.
949     a. fragment : 프래그먼트를 식별하기 위한 이름
950     b. var : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.lang.String 타입으로 저장. var와
        varReader 함께 사용 못함.
951     c. varReader : 태그 바디 실행 결과를 담을 유효범위 속성의 이름. 값은 java.io.Reader 타입으로 저장. var와
        varReader 함께 사용 못함.
952     d. scope : 결과 값의 유효범위
953
954 *****invokeTest.jsp*****
955 <%@ page language="java" contentType="text/html; charset=UTF-8"
956     pageEncoding="UTF-8"%>
957 <%@ taglib prefix="easy" tagdir="/WEB-INF/tags" %>
958 <!DOCTYPE html>
959 <html lang="ko">
960 <head>
961 <meta charset="UTF-8">
962 <title>Invoke Demo</title>
963 </head>
964 <body>
965     <easy:invokeDemo>
966         <jsp:attribute name="productDetails">
967             <table width="220" border="1">
968                 <tr>
969                     <td><b>상품명</b></td>
970                     <td>${productName}</td>
971                 </tr>
972                 <tr>
973                     <td><b>설명</b></td>
974                     <td>${description}</td>
975                 </tr>
976                 <tr>
977                     <td><b>가격</b></td>
978                     <td>${price}</td>
979                 </tr>
980             </table>
981         </jsp:attribute>
982     </easy:invokeDemo>
983 </body>
984 </html>
985 *****invokeDemo.tag*****
986 <%@ tag pageEncoding="utf-8" %>
987 <%@ attribute name="productDetails" fragment="true" %>
988 <%@ variable name-given="productName" %>
989 <%@ variable name-given="description" %>
990 <%@ variable name-given="price" %>
991 <%

```

```
992     jspContext.setAttribute("productName", "오성전자 DVD 플레이어");
993     jspContext.setAttribute("description", "동축 디지털 오디오 잭을 통한 돌비 디지털 출력, " +
994         "풀 HD 화질 재생");
995     jspContext.setAttribute("price", "65");
996 %>
997 <jsp:invoke fragment="productDetails" />
```