

```
1 1. JSTL?
2 1)표준화된 tag library들을 제공함으로써 보다 편리하게 Web application을 개발할 수 있도록 지원
3 2)JSTL 1.1 은 Servlet 2.4 이상, JSP 2.0 이상
4 3)JSTL 1.2 은 Servlet 2.5 이상, JSP 2.1 이상
5 4)Java Community Process(www.jcp.org)의 JSR-52 가 관리
6
7
8 2. Homepage
9 1)http://tomcat.apache.org/taglibs/standard/
10 2)https://www.oracle.com/technetwork/java/jstl-137486.html
11
12
13 3. Downloads
14 1)JSTL을 사용하려면 JSTL API 와 JSTL 구현체를 모두 다운로드해야 함.
15 2)JSTL 1.1.x
16 -http://archive.apache.org/dist/jakarta/taglibs/standard/binaries/
17 -downloads jakarta-taglibs-standard-1.1.2.zip
18 -Unzip
19 -/lib/jstl.jar, /lib/standard.jar
20 -Copy to : /WEB-INF/lib
21
22 3)JSTL 1.2.x
23 -http://tomcat.apache.org/download-taglibs.cgi
24 -Downloads
25 --Impl : taglibs-standard-impl-1.2.5.jar
26 --Spec : taglibs-standard-spec-1.2.5.jar
27 -Copy to : /WEB-INF/lib
28
29
30 4. JSTL Example Sites
31 1)Eclipse에서 standard-doc.war 과 standard-examples.war file을 import한다.
32 -import -> WAR files -> standard-examples.war 선택
33 2)http://localhost:8080/standard-examples/index.html
34
35
36 5. JSTL 에서 제공하는 Tag의 종류
37 1)http://jakarta.apache.org/taglibs/doc/standard-doc/standard/GettingStarted.html 참조하자.
38 2)Core
39 -http://java.sun.com/jsp/jstl/core
40 -c
41 -<c:tagname...>
42
43 3)XML processing
44 -http://java.sun.com/jsp/jstl/xml
45 -x
46 -<x:tagname...>
47
48 4)I18N capable formatting(국제화)
49 -http://java.sun.com/jsp/jstl/fmt
50 -fmt
51 -<fmt:tagname...>
52
53 5)Database access(SQL)
54 -http://java.sun.com/jsp/jstl/sql
55 -sql
56 -<sql:tagname...>
57
58 6)Functions
59 -http://java.sun.com/jsp/jstl/functions
60 -fn
61 -fn:functionName(...)
62
63
64 6. JavaServer Pages Standard Tag Library 1.1 Tag Reference
65 -http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html
66 -http://docs.oracle.com/cd/E17802\_01/products/products/jsp/jstl/1.1/docs/tlddocs/index.html
67
```

```

68
69 7. JavaServer Pages Standard Tag Library
70   -https://docs.oracle.com/javaee/7/index.html
71
72
73 8. Core Library
74   1)JSP page에서 필요한 가장 핵심적이고 기본적인 기능들을 제공
75   2)<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
76   3)coredemo.jsp
77       <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
78       <c:forEach begin="1" end="10">
79           <h3>Hello, World</h3>
80       </c:forEach>
81
82   4)Tags
83       -c:catch : 예외처리에 사용
84       -c:out : JspWriter 에 내용 출력
85       -c:set : JSP 에서 사용될 변수 설정, setAttribute(name, key) 기능과 동일, scope 에 따른 바인딩 처리 가능
86       -c:remove : 설정한 변수 제거, removeAttribute(name) 과 동일, scope 에 따른 속성 제거 가능
87       -c:if : 조건처리
88       -c:choose : 다중 조건 처리
89       -c:forEach : 컬렉션이나 Map 의 각 항목 처리
90       -c:forTokens : 구분자로 분리된 각각의 토큰을 처리할 때 사용
91       -c:when
92       -c:otherwise
93       -c:import : URL 을 사용하여 다른 자원의 결과를 삽입
94       -c:redirect : URL 을 재작성한다.
95       -c:param
96
97   5)범례
98       -required : 필수
99       -rtexprvalue = true : 정적문자열 및 동적인 값(자바표현식, EL 표현식, <jsp:attribute />)로 설정값 모두
        가능함을 의미
        false : 정적 문자열만 할당가능
100
101
102   6)<c:out>
103       -지정된 값 출력
104       -<c:out value="출력값" [default="기본값"] [escapeXml="{true | false}"] />
105       -<c:out value="출력값" [default="기본값"] [escapeXml="{true | false}"]>
106           default value
107       </c:out>
108       -value(rtexprvalue = true, required)
109           --변수의 내용 출력
110       -default(rtexprvalue = true)
111           --기본 값을 의미
112           --value 값이 null일 경우 이 기본 값 출력
113       -escapeXml(rtexprvalue = true)
114           --false가 기본으로 설정
115           --<, > 등의 특수 기호를 처리할 때 사용
116           --true 이면 < 값은 &lt; 으로 표현, > 값은 &gt; 로 표현
117
118   [outdemo.jsp]
119   <%@ page import = "java.io.IOException, java.io.FileReader" %>
120   <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
121   <html>
122       <head><title>소스 보기</title></head>
123       <body>
124       <c:catch var="err">
125       <%
126           FileReader reader = null;
127           String path = request.getParameter("path");
128           reader = new FileReader(getServletContext().getRealPath(path));
129       %>
130       <pre>
131           소스 코드 = <%= path %>
132           <c:out value="<%= reader %>" escapeXml="true" />
133       </pre>

```

```

134         </c:catch>
135         <h3>Message : <c:out value="\${err}"/></h3>
136     </body>
137 </html>
138 http://localhost .... /outdemo.jsp?path=setdemo.jsp
139
140 [outdemo1.jsp]
141 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
142 <c:out value="<font size='7'>커다란 글씨</font>는 다음과 같이 출력합니다" /><br /><br />
143 <c:out value="<font size='7'>커다란 글씨</font>는 다음과 같이 출력합니다" escapeXml = "false" />
144

```

```

145 [outdemo2.jsp]
146 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
147 안녕하세요. <c:out value="\${param.id}" default="guest" />님!
148

```

7) <c:set>

```

150 -지정된 변수에 값을 설정
151 -<c:set var="변수명" value="설정값" target="객체" property="값" [scope="범위"] />
152 -var : 값을 저장할 변수의 이름
153 -value(rtexprvalue = true) : 저장할 값
154 -target(rtexprvalue = true) : 값을 설정할 property에 대한 객체(bean객체, Map객체)를 의미
155 -property(rtexprvalue = true) : 값을 설정할 객체의 property
156 -scope : 변수의 유효범위, 설정하지 않으면 기본값으로 page
157

```

```

158 [setdemo.jsp]
159 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
160 <body>
161     <c:set var="browser" value="\${header['user-agent']}" />
162     삭제전 <br />
163     <c:out value="\${browser}" /><br />
164     <c:remove var="browser"/>
165     삭제후<br />
166     <c:out value="\${browser}" />
167 </body>
168

```

```

169 [setdemo1.jsp]
170 <%@ page import="java.util.Date" %>
171 <%@ page import="java.util.HashMap" %>
172 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
173 <c:set var="num1" value="\${20}"/>
174 <c:set var="num2" >
175     10.5
176 </c:set>
177 변수 num1 : ${num1}<br />
178 변수 num2 : ${num2}<br />
179 num1 + num2 = ${num1 + num2}<br />
180
181 <c:set var="today" value="<%=new Date() %>" />
182 Today is ${today}.
183 <br />
184 <c:set var="map" value="<%=new HashMap()%>" />
185 <c:set target="\${map}" property="irum" value="조성모" />
186 귀하의 이름은 ${map.irum} 이군요.
187

```

```

188 [LoginBean.java]
189 package com.example.libs;
190
191 public class LoginBean {
192     private String userid;
193     private String passwd;
194
195     public String getUserid() {
196         return userid;
197     }
198     public void setUserid(String userid) {
199         this.userid = userid;
200     }

```

```

201     public String getPasswd() {
202         return passwd;
203     }
204     public void setPasswd(String passwd) {
205         this.passwd = passwd;
206     }
207 }
208
209 [setdemo2.jsp]
210 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
211 <jsp:useBean id="myBean" class="com.example.libs.LoginBean" />
212
213 <c:set target="${myBean}" property="userid" value="example" />
214 ${myBean.userid} <br />
215 <c:out value="${myBean.userid}" />
216
217 [setdemo3.jsp]
218 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
219 <c:set var="CODE" value="80012" scope="request" />
220 <c:set var="NAME" value="Pencil" scope="request" />
221 <c:set var="PRICE" value="1000" scope="request" />
222 <jsp:forward page="productview.jsp" />
223
224 [productview.jsp]
225 <h3>Product Information</h3>
226 상품코드 : ${CODE}<br />
227 상품명 : ${NAME}<br />
228 단가 : ${PRICE}원
229
230 [com.example.LoginBean.java]
231 public class LoginBean {
232     private String userid;
233     private boolean verified;
234
235     public String getUserid() {
236         return userid;
237     }
238     public void setUserid(String userid) {
239         this.userid = userid;
240     }
241     public boolean isVerified() {
242         return verified;
243     }
244     public void setVerified(boolean verified) {
245         this.verified = verified;
246     }
247 }
248
249 [login.html]
250 <!DOCTYPE html>
251 <html>
252 <head>
253 <meta charset="UTF-8">
254 <title>로그인해 주세요.</title>
255 </head>
256 <body>
257     <form action="login.jsp" method="post">
258         ID : <input type="text" name="userid" /><br />
259         PWD : <input type="password" name="userpwd" /><br />
260         <input type="submit" value="로그인" />
261     </form>
262 </body>
263 </html>
264
265 [login.jsp]
266 <%@ page language="java" contentType="text/html; charset=UTF-8"
267     pageEncoding="UTF-8"%>

```

```

268 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
269
270 <jsp:useBean id="login" class="com.example.LoginBean" />
271
272 <c:if test="${param.userid eq 'javaexpert' and param.userpwd eq '12345678'}">
273     <jsp:setProperty property="userid" name="login" value="${param.userid}"/>
274     <jsp:setProperty property="verified" name="login" value="true"/>
275     <c:set var="userid" value="${param.userid}" scope="session" />
276     <c:set var="verified" value="true" scope="session" />
277     <c:out value="<div style='color:green'>Login Success</div>" escapeXml="false" />
278     <a href="secure.jsp">회원전용페이지로...</a>
279 </c:if>
280 <c:if test="${param.userid ne 'javaexpert' or param.userpwd ne '12345678'}">
281     <script>
282         alert("아이디 혹은 비밀번호가 일치하지 않습니다.");
283         history.back();
284     </script>
285 </c:if>

```

```

286
287 [secure.jsp]
288 <%@ page language="java" contentType="text/html; charset=UTF-8"
289     pageEncoding="UTF-8"%>
290 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
291
292 <c:set var="verified" value="${sessionScope.verified}" />
293 <c:set var="userid" value="${sessionScope.userid}" />
294
295 <c:if test="${verified ne 'true'}">
296     <script>
297         alert("먼저 로그인부터 해주세요.");
298         location.href = 'login.html';
299     </script>
300 </c:if>
301
302 <c:if test="${verified eq 'true'}">
303     <script>
304         var str = "Welcome!!!<c:out value='${userid}' />님!!!";
305         alert(str);
306     </script>
307 </c:if>

```

8) <c:remove>

```

310 -설정된 값 제거하는 tag
311 -<c:remove var="변수명" [scope="범위"] />
312 -var : 설정된 속성이 저장되어 있는 변수
313 -scope : 지정된 범위에서 지정된 속성을 제거
314

```

```

315 [removedemo.jsp]
316 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
317 <body>
318 <c:set var="test" value="Hello JSTL!"/>
319 <h3>&lt;c:set&gt; 사용 후 : <c:out value="${test}"/></h3>
320 <c:remove var="test"/>
321 <h3>&lt;c:remove&gt; 사용 후 : <c:out value="${test}"/></h3>
322 </body>
323

```

9) <c:catch>

```

325 -예외 처리를 위한 tag
326 -<c:catch var="변수명" />
327 -예외가 발생하면 var(rtexprvalue = true) 에 지정된 예외 객체가 할당
328

```

```

329 [catchdemo.jsp]
330 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
331 <c:catch var="err">
332     <%=10/0%>
333 </c:catch>
334 <h3>Message : <c:out value="${err}"/></h3>

```

```

[catchdemo1.jsp]
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:catch var="ex">
    name 파라미터의 값 = <%= request.getParameter("name") %> <br>
    <% if (request.getParameter("name").equals("test")) { %>
        ${param.name}은 test 입니다.
    <% } %>
</c:catch>
<p>
<c:if test="${ex != null}">
    예외가 발생하였습니다:<br>
    ${ex}
</c:if>

```

10)<c:if>

- 조건 처리할 때 사용되는 tag
- <c:if test="조건" var="변수명" [scope="범위"] />
- test(rtexprvalue = true) : 조건을 지정
- var : 속성의 변수에는 조건 처리할 결과를 저장
- scope : var 속성에 지정한 변수의 범위를 의미

[ifdemo.jsp]

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:if test="true">
    무조건 수행된다.
</c:if>
<br />

<c:if test="${5<10}">
<h3>5는 10보다 작다.</h3>
</c:if>
<c:if test="${6+3==9}">
<h3>6 + 3 은 9이다.</h3>
</c:if>
<br />

<c:if test="${param.name eq 'code'}" >
    name 파라미터의 값은 ${param.name} 입니다.
</c:if>
<br />

<c:if test="${!empty param.name}" >
    Name : ${param.name}
</c:if>
<c:if test="${param.age > 18}" >
    귀하는 성인이지군요.
</c:if>

```

11)<c:choose>

- 조건 처리할 때 사용되는 tag
- <c:choose>
- <c:when test="조건"></c:when>
- <c:otherwise></c:otherwise>
- </c:choose>
- Java의 switch 와 비슷
- <c:when> tag의 test 속성으로 조건 확인하며, 조건에 만족하면
- <c:when> 과 </c:when> 사이에 있는 내용 처리
- <c:when> tag는 조건에 따라 여러 개 사용가능
- 어느 조건도 만족하지 않을 때 <c:otherwise> 실행

[choosedemo.jsp]

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:choose>
    <c:when test="${5+10==50}">
        <h3>5+10은 50이다.</h3>
    </c:when>

```

```

402
403     <c:otherwise>
404         <h3>5+10은 50이 아니다.</h3>
405     </c:otherwise>
406 </c:choose>
407
408 [choosedemo1.jsp]
409 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
410 <body>
411 <c:choose>
412     <c:when test="${param.name == 'code'}">
413         당신의 이름은 ${param.name} 입니다.<br>
414     </c:when>
415     <c:when test="${param.age > 18}" >
416         당신은 미성년자가 아닙니다.
417     </c:when>
418     <c:otherwise>
419         당신은 code도 아니고 미성년자입니다.
420     </c:otherwise>
421 </c:choose>
422 </body>
423
424 [season.html]
425 <!DOCTYPE html>
426 <html>
427 <head>
428 <meta charset="UTF-8">
429 <title>Insert title here</title>
430 </head>
431 <body>
432     <form action="choosedemo.jsp" method="post">
433         당신이 좋아하시는 계절은 ?
434         <input type="text" name="season">
435         <input type="submit" value="전송" />
436     </form>
437 </body>
438 </html>
439
440 [choosedemo.jsp]
441 <%@ page language="java" contentType="text/html; charset=UTF-8"
442     pageEncoding="UTF-8"%>
443 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
444
445 <c:set var="myseason" value="${param.season}" />
446
447 <%-- <c:if test="${myseason eq 'spring'}">
448     <h2>개나리, 진달래</h2>
449 </c:if>
450 <c:if test="${myseason eq 'summer'}">
451     <h2>장미, 아카시아</h2>
452 </c:if>
453 <c:if test="${myseason eq 'fall'}">
454     <h2>백합, 코스모스</h2>
455 </c:if>
456 <c:if test="${myseason eq 'winter'}">
457     <h2>동백, 매화</h2>
458 </c:if> --%>
459
460 <c:choose>
461     <c:when test="${myseason eq 'spring'}">
462         <h2>개나리, 진달래</h2>
463     </c:when>
464     <c:when test="${myseason eq 'summer'}">
465         <h2>장미, 아카시아</h2>
466     </c:when>
467     <c:when test="${myseason eq 'fall'}">
468         <h2>백합, 코스모스</h2>

```

```

469     </c:when>
470     <c:when test="\${myseason eq 'winter'}">
471         <h2>동백, 매화</h2>
472     </c:when>
473     <c:otherwise>
474         <h2 style='color:red'>잘못된 계절을 입력하셨습니다.</h2>
475     </c:otherwise>
476 </c:choose>
477

```

12) <c:forEach>

```

479 -반복 처리할 때 사용되는 tag, Java의 for문과 유사
480 -<c:forEach begin="시작 index" end="끝 index" step="증감식" [var="변수명"] /> or
481 <c:forEach items="객체명" [begin="시작 index"] [end="끝 index"]
482     [step="증감식"] [var="변수명"] [varStatus="상태변수"] />
483 -items(rtexprvalue = true) : collection이나 배열 형태의 객체를 지정하여 객체의 index만큼 반복 가능
484 -begin(rtexprvalue = true) 과 end(rtexprvalue = true) 속성으로 원하는 범위만큼 반복 수행도 가능
485 -step(rtexprvalue = true) : 증감식 설정
486 -var : 반복 중일 때 현재 반복하는 있는 값을 기억하는 변수
487 -varStatus : 반복의 상태를 갖는 변수
488

```

[foreachdemo.jsp]

```

490 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
491 <c:forEach var="test" begin="1" end="10" step="2">
492     <b>\${test}</b>&nbsp;
493 </c:forEach>
494 <br>
495

```

```

496 <h2>1부터 100까지의 홀수의 합</h2>
497 <c:set var="sum" value="0" />
498 <c:forEach var="i" begin="1" end="100" step="2">
499     <c:set var="sum" value="\${sum + i}" />
500 </c:forEach>
501 결과 : \${sum}<br />
502

```

```

503 <h2>구구단</h2>
504 <table cellpadding="0" cellspacing="0" border="1" width="800">
505     <tr height="30">
506         <c:forEach var="j" begin="2" end="9">
507             <th width="100">\${j}</th>
508         </c:forEach>
509     </tr>
510     <c:forEach var="i" begin="1" end="9">
511         <tr height="20">
512             <c:forEach var="j" begin="2" end="9">
513                 <td width="100">
514                     \${j} x \${i} = \${j * i}
515                 </td>
516             </c:forEach>
517         </tr>
518     </c:forEach>
519 </table>
520

```

[foreachdemo1.jsp]

```

522 <%@ page import="java.util.HashMap" %>
523 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
524 <%
525     HashMap mapObj = new HashMap();
526     mapObj.put("name", "조성모");
527     mapObj.put("today", new java.util.Date());
528 %>
529 <c:set var="intArray" value="\<%= new int[] {1,2,3,4,5} %>" />
530 <h4>int형 배열</h4>
531
532 <c:forEach var="i" items="\${intArray}" begin="2" end="4">
533     [\${i}]
534 </c:forEach>
535 <c:set var="map" value="\<%=mapObj %>" />

```



```

536 </h2>Map</h2>
537 <c:forEach var="i" items="${map}">
538     ${i.key} = ${i.value}<br />
539 </c:forEach>
540
541
542 [foreachdemo2.jsp]
543 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
544 <body>
545     <h1>Header 정보</h1>
546     <c:forEach var="head" items="${headerValues}">
547         param : <c:out value="${head.key}" /><br />
548         value :
549         <c:forEach var="value" items="${head.value}">
550             <c:out value="${value}" />
551         </c:forEach><br />
552     </c:forEach>
553 </body>
554
555 [foreachdemo3.jsp]
556 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
557 <c:forEach var="cnt" begin="1" end="7">
558     <font size="${cnt}">Hello, World</font><br />
559 </c:forEach>
560

```

13)<c:forTokens>

-Java의 for문과 StringTokenizer 객체를 결합한 것 같음.
 -<c:forTokens items="객체명" delims="구분자" [begin="시작 인덱스"] [end="끝 인덱스"] [step="증감식"] [var="변수명"] [varStatus="상태변수"] />
 -items(rtexprvalue = true) : 지정된 값을 delims(rtexprvalue = true) 속성의 구분자로 나눈 후 나뉜 만큼 반복을 수행
 -items, delims, var 이 세가지의 속성만으로도 <c:forTokens> tag 수행 가능
 -begin(rtexprvalue = true)과 end(rtexprvalue = true)의 경우는 delims 구분자로 나뉜 값들을 기준으로 시작 값과 끝 값을 정해주기에 가능

```

568
569 [fortokensdemo.jsp]
570 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
571 <%-- <%
572     String str = "C++, Java, Visual C#, Python, R, Ruby";
573     StringTokenizer st = new StringTokenizer(str, ",");
574     int count = st.countTokens();
575     for(int i = 0 ; i < count ; i++){
576         out.print(st.nextElement() + "&nbsp;&nbsp;&nbsp;");
577     }
578 %> --%>
579
580 <c:set var="mytokens" value="C++, Java, Visual C#, Python, R, Ruby" />
581 <c:forTokens var="i" items="${mytokens}" delims=",">
582     ${i}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
583 </c:forTokens>
584
585 [fortokensdemo1.jsp]
586 콤마와 점을 구분자로 사용:<br>
587 <c:forTokens var="token"
588     items="빨강색,주황색,노란색,초록색,파랑색,남색,보라색"
589     delims=",">
590     ${token}
591 </c:forTokens>
592

```

14)<c:import>

-지정된 URL tag가 사용된 JSP page에 출력시키는 기능
 -<jsp:include> 와 비슷
 -<c:import url="URL 값" var="변수명" scope="범위" varReader="입력스트림명" context="contextName" charEncoding="인코딩값" />
 -url : HTTP 뿐만 아니라 FTP 외부 resource도 가능
 -var : resource가 저장될 변수
 -scope : var 속성에 지정한 변수의 범위를 의미

-varReader : var 속성과 마찬가지로 resource가 저장될 변수를 의미. 단, 이 변수는 Reader 객체이어야 한다.
-context : URL에 접근할 때의 context 이름을 의미
-charEncoding : 지정된 URL의 resource를 가져올 때 어떻게 encoding할 것인지를 의미

[importdemo.jsp]

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:import url="/fortokendsdemo.jsp" var="myurl"></c:import>
<body>
  <h1>Import 연습하기</h1>
  ${myurl}
</body>
```

[footer.jsp]

```
<%@ page contentType = "text/html; charset=utf-8" %>
<hr size="1">
<center>footer area</center>
```

[header.jsp]

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<hr size="1">
<c:if test="${param.id != null}">
[HEADER] ${param.id} 님의 방문을 환영합니다.
<hr size="1">
</c:if>
```

[importdemo1.jsp]

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:import url="setdemo.jsp" var="urlEx" />
<c:import url="header.jsp" var="head">
  <c:param name="id" value="javaexpert" />
</c:import>
<c:import url="footer.jsp" var="foot" />

<html>
<head><title>import와 url tag</title></head>
<body>

  ${head}

  ${urlEx}

  ${foot}

</body>
</html>
```

15)<c:redirect>

-지정된 URL로 page를 이동시키는 기능
-<c:redirect url="URL 값" context="contextName" />
-url : 이동할 URL 입력
-context : context 이름을 입력하지만, 일반적으로 url 속성만 사용

[redirectdemo.jsp]

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:redirect url="multiply.jsp">
  <c:param name="num1" value="5" />
  <c:param name="num2" value="25" />
</c:redirect>
```

16)<c:url>

-value 속성에 지정된 값으로 URL 을 생성하는 기능
-<c:url var="변수명" scope="범위" value="값" context="contextName" />
-scope : var 속성에 입력되어 있는 변수의 범위를 의미

[urldemo.jsp]

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```

668 <c:url value="multiply.jsp" var="abc">
669     <c:param name="num1" value="19" />
670     <c:param name="num2" value="25" />
671 </c:url>
672 <c:redirect url="\${abc}" />
673
674 [multiply.jsp]
675 <%@ page language="java" contentType="text/html; charset=UTF-8"
676     pageEncoding="UTF-8"%>
677 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
678
679 <c:set var="first" value="\${param.first}" />
680 <c:set var="second" value="\${param.second}" />
681
682 <h1>multiply.jsp</h1>
683 \${first} x \${second} = \${first * second}
684

```

17) <c:param>

```

686 -<c:import> tag에 parameter를 전달하기 위한 tag
687 -<c:param name="파라미터명" value="값" />
688 -name : parameter명을 입력
689 -value : 값을 입력하여 전달
690

```

9. I18N capable formatting(국제화)

1) 국제화/형식화의 기능을 제공해 주는 library

2) 국제화는 다국어 내용을 처리, 형식화는 날짜와 숫자 형식 등을 처리

3) <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

4) <fmt:requestEncoding>

```

697 -request 객체로부터 전달 받은 값들을 encoding할 때 사용
698 -<frm:requestEncoding value="encoding값" />
699 -request.setCharacterEncoding와 같다.
700

```

[fmtdemo.jsp]

```

702 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
703 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
704 <fmt:requestEncoding value="utf-8" />
705 <%--
706     <% request.setCharacterEncoding("utf-8"); %>
707 --%>
708
709 <body>
710     <form method="post">
711         Name : <input type="text" name="irum"/>
712         <input type="submit" value="전송" />
713     </form><br />
714     Name : <c:out value="\${param.irum}"></c:out>
715 </body>
716 </html>
717

```

5) <fmt:setLocale>

```

719 -다국어 page를 사용할 때 언어를 지정하는 tag
720 -<frm:setLocale value="값" variant="" scope="범위" />
721 -여러 나라에 해당하는 언어를 지원하도록 설계되어 있을 때 어떤 언어를 사용하여 page를 표시할지 설정하는 tag
722 -value : 어떤 언어를 사용할 지 지정하는 속성이며 언어 code를 입력
723 -variant : Browser의 spec을 지정할 때 사용
724

```

6) <fmt:timeZone>

```

726 -지정한 지역 값으로 시간대를 맞출 때 사용
727 -<fmt:timeZone value="값" />
728 -<fmt:setTimeZone>과 비슷하지만, <fmt:setTimeZone>은 특정 page 전체에 적용
729 -하지만, 이 tag는 첫 tag와 끝 tag 사이의 body 부분의 내용에만 적용
730 -java.util.TimeZone 값 지정 가능
731

```

```

732 <fmt:timeZone value="GMT+1:00">
733     <fmt:formatDate value="\${now}" type="both" dateStyle="full"
734         timeStyle="full" />

```

```

735     </fmt:timeZone>
736
737 7)<fmt:setTimeZone>
738     -위와 비슷
739     -<fmt:setTimeZone value="값" [var="변수명"] [scope="범위"] />
740     -var : 변수에 지정한 시간대 값이 저장
741     -scope : 범위 지정
742
743 8)<fmt:bundle>
744     -properties 확장명을 가진 file의 resource를 불러올 때 사용
745     -<fmt:bundle basename="basename" prefix="prefix" />
746     -보통 다국어 처리 때 사용
747     -basename : properties 확장명을 가진 file을 지정
748     -Package까지 포함한 경로를 지정해야
749     -prefix : properties 내의 key 값에 쉽게 접근할 수 있도록 간략한 접근어를 사용
750     -불러온 resource는 이 tag의 body 내에서만 사용가능
751
752 9)<fmt:setBundle>
753     -위와 같은 기능, 하지만, 사용영역의 차이다.
754     -<fmt:setBundle basename="basename" var="변수명" scope="범위" />
755     -<fmt:bundle> tag는 첫 tag와 끝 tag 사이의 body 영역에서만 불러온 resource를 사용하지만,
756     -이 tag는 page 전체에서 사용
757     -var : 설정한 resource 내용을 가지고 있을 변수명
758     -scope : 범위 지정
759
760 10)<fmt:message>
761     -설정한 properties file의 resource 내용을 읽을 때 사용
762     -<fmt:message key="키값" bundle="bundle변수" var="변수명" scope="범위" />
763     -properties file에는 여러 key값들로 내용이 구분된다.
764     -key 속성값으로 설정된 내용을 가져올 수 있다.
765     -bundle : <fmt:setBundle>의 var에서 지정했던 변수를 입력
766     -이 tag는 bundle 속성을 사용하여 properties file에 접근하는 것임.
767
768     /WEB-INF/classes folder에 생성, EE Eclipse를 사용할 경우 src folder에 저장하고 한글로 입력하면 자동으로
769     ASCII로 변환
770     [message_en.properties]
771         TITLE=Welcome to example.com
772         GREETING=Welcome!^^
773         VISITOR=Your ID is {0}
774
775     [message_ko.properties]
776         TITLE=\uD658\uC601\uD569\uB2C8\uB2E4.
777         GREETING=\uC548\uB155\uD558\uC138\uC694.
778         VISITOR=\uB2F9\uC2E0\uC758 \uC57C\uC774\uB514\uB294 {0}
779         -----
780         TITLE=환영합니다.
781         GREETING=안녕하세요.
782         VISITOR=당신의 아이디는 {0} 입니다.
783     -한글로 입력하면 자동으로 변환됨
784
785 [frmdemo1.jsp]
786 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
787 <fmt:setBundle basename="message_ko" var="mymsg" />
788 <fmt:message key="TITLE" var="title" bundle="\${mymsg}"/>
789 <!DOCTYPE html>
790 <html>
791 <head>
792 <meta charset="UTF-8">
793 <title>\${title}</title>
794 </head>
795 <body>
796 <div><fmt:message key="GREETING" bundle="\${mymsg}"/></div>
797 <div><fmt:message key="VISITOR" bundle="\${mymsg}">
798     <fmt:param value="\${param.id}" />
799 </fmt:message></div>
800 </body>
</html>

```

11)<fmt:param>

- <fmt:message> tag로 읽어 온 resource 내용에 parameter를 전달하는 tag
- resource 내용이 만일 '{0}'님 안녕하세요'일 경우 <fmt:param> tag로 {0} 대신 원하는 parameter 지정

12)<fmt:formatNumber>

- 숫자 형식의 pattern 설정할 때 사용
- <fmt:formatNumber value="값" [type="{number | currency | percent}"] [pattern="패턴"] [currencyCode="값"] [currencySymbol="값"] [groupingUsed="{true | false}"] [maxIntegerDigits="값"] [minIntegerDigits="값"] [maxFractionDigits="값"] [minFractionDigits="값"] [var="변수명"] [scope="범위"] />
- value(rtexprvalue = true) : pattern을 적용시킬 원래의 값
- type(rtexprvalue = true) : 숫자의 type, type은 number, currency, percent 중 원하는 type 설정
- pattern(rtexprvalue = true) : 지정한 값을 어떤 pattern으로 변화시킬지 결정
- currencyCode(rtexprvalue = true) : 통화 code를 의미, 숫자 type이 통화일 경우만 유효

-<ISO 4217의 통화 코드 지정>

캐나다 달러	CAD	달러	센트
중국 위안	CNY	위안	자오
유로	EUR	유로	유로 센트
일본 엔	JPY	엔	엔
영국 파운드	GBP	파운드	펜스
미국 달러	USD	달러	센트

- currencySymbol(rtexprvalue = true) 도 숫자 type이 통화일때만 유효, 통화 기호 지정
- groupingUsed(rtexprvalue = true) : 그룹 기호의 포함 여부
- maxIntegerDigits(rtexprvalue = true) : 최대 정수 길이
- minIntegerDigits(rtexprvalue = true) : 최소 정수 길이
- maxFractionDigits(rtexprvalue = true) : 최대 소수점 자릿수
- minFractionDigits(rtexprvalue = true) : 최소 소수점 자릿수

ex)

```
<...value="12" type="number" /> --> 12
<...value="12" type="number" minIntegerDigits="3" /> --> 012
<...value="12" type="number" minFractionDigits="2" /> --> 12.00
<...value="123456.78" pattern=".000" /> --> 123456.780
<...value="123456.78" pattern="#,#00.0#" /> --> 123, 456.78
<...value="12" type="currency" /> --> $12.00
<...value="12" type="currency" currencyCode="GBP" /> --> GBP 12.00
<...value="0.12" type="percent" /> --> 12%
<...value="0.125" type="percent" minFractionDigits="2" /> --> 12.50%
```

[frmdemo2.jsp]

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
Number : <fmt:formatNumber value="12345678.3456" type="number" /><br />
Currency : <fmt:formatNumber value="1500000"
           type="currency" currencySymbol="₩"/><br />
Currency : <fmt:formatNumber value="1500000"
           type="currency" currencySymbol="$"/><br />
Currency : <fmt:formatNumber value="1500000"
           type="currency" currencySymbol="¥"/><br />
Percent : <fmt:formatNumber value="1500000" type="percent" /><br />
<c:set var="now" value="<%=new java.util.Date()%>" />
Now is ${now}<br />
Now is <c:out value="${now}" /><br />
Date : <fmt:formatDate value="${now}" type="date" /><br />
Time : <fmt:formatDate value="${now}" type="time" /><br />
Both : <fmt:formatDate value="${now}" type="both" /><br />

<fmt:formatNumber value="50000" type="currency"/><br>
<fmt:formatNumber value="0.15" type="percent"/><br>
<fmt:formatNumber value="500567300" pattern="###,###,###"/><p>

<jsp:useBean id="date" class="java.util.Date"/>
<fmt:formatDate value="${date}" type="date"/><br>
<fmt:formatDate value="${date}" type="time"/><br>
```

```

868 <fmt:formatDate value="\${date}" type="both"/><p>
869
870 <fmt:formatDate value="\${date}" type="both" timeStyle="short" dateStyle="short"/><br>
871 <fmt:formatDate value="\${date}" type="both" timeStyle="long" dateStyle="long"/><br>
872

```

13)<fmt:parseNumber>

```

874 -문자열을 숫자, 통화 또는 percent의 형태로 변환할 때 사용하는 tag
875 -<fmt:parseNumber value="값" [type="{number | currency | percent}"]
876 [pattern="패턴"] [parseLocale="값"] [integerOnly="{true | false}"]
877 [var="변수명"] [scope="범위"] />
878 -value 속성에 문자열 값
879 -type 속성에는 숫자, 통화, percent 중 변환할 타입 설정
880 -pattern 속성에는 value 속성의 값을 설정된 type으로 변환하면서 어떤 pattern을 갖게 할 것인지 지정
881 -parseLocale 은 parsing할 기본 pattern 지정
882 -integerOnly 는 지정된 값 중 정수 부분만 해석할지 여부 결정
883

```

14)<fmt:formatDate>

```

885 -날짜 형식의 pattern을 설정할 때 사용되는 tag
886 -<fmt:formatDate value="값" [type="{time | date | both}"]
887 [dateStyle="{ default | short | medium | long | full }"]
888 [timeStyle="{ default | short | medium | long | full }"]
889 [pattern="패턴"] [timeZone="값"] [var="변수명"] [scope="범위"] />
890 -value(rtexprvalue = true) 속성에는 날짜 또는 시간을 입력
891 -type(rtexprvalue = true) 속성으로 날짜인지 시간인지 또는 날짜 시간 둘다 포함한 type인지 지정
892 -dateStyle(rtexprvalue = true) 은 날짜 style 지정, type 속성이 date 또는 both 일 때만
893 -timeStyle(rtexprvalue = true) 은 시간 style 지정, type 속성이 time 또는 both 일 때만
894 -timeZone(rtexprvalue = true) 은 날짜와 시간이 표시될 시간대 지정
895

```

ex)

```

897 -기본format : <fmt:formatDate value="\${now}" />
898 -짧은format : <fmt:formatDate value="\${now}" dateStyle="short" />
899 -중간format : <fmt:formatDate value="\${now}" dateStyle="medium"/>
900 -긴format : <fmt:formatDate value="\${now}" dateStyle="long" />
901 -전체format : <fmt:formatDate value="\${now}" dateStyle="full" />
902 - 시간 format은 기본적으로 위의 날짜 format과 같음.
903

```

[fmtdemo2.jsp]

```

905 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
906 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
907
908 <body>
909 <c:set var="now" value="<%=new java.util.Date() %>"/>
910 KOREA Time : <fmt:formatDate value="\${now}"
911 type="both" dateStyle="full" timeStyle="full" /><br />
912 UK Time : <fmt:timeZone value="GMT">
913 <fmt:formatDate value="\${now}"
914 type="both" dateStyle="full" timeStyle="full" />
915 </fmt:timeZone>
916 </body>
917

```

15)<fmt:parseDate>

```

919 -문자열을 날짜와 시간의 형태로 변환하는 tag
920 -<fmt:parseDate value="값" [type="{time | date | both}"]
921 [dateStyle="{ default | short | medium | long | full }"]
922 [timeStyle="{ default | short | medium | long | full }"]
923 [pattern="패턴"]
924 [timeZone="값"]
925 [parseLocale="값"] [var="변수명"] [scope="범위"] />
926 -value 속성에 입력된 문자열 값을 type 속성에 지정된 타입으로 날짜와 시간 형태로 변환
927
928 <c:set var="myDate" value="12/12/2005" />
929 <fmt:parseDate var="formattedDate" type="date" dateStyle="short"
930 value="\${myDate}" />
931

```

10. JSTL sql

1)SQL 관련 기능을 제공해 주는 JSTL library

2) Database server에 접근, query 전송, transaction 처리 가능

3) <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

4) <sql:setDataSource />

- Database에 접근
- <sql:setDataSource var="variableName" scope="{page | request | session | application}" {dataSource="dataSource" | driver="driver" url="url" user="user" password="password"} />
- 접근방법 2가지
 - a. dataSource
 - <sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
 - dataSource 가 JNDI 로 설정되었으면 바로 <sql:query> 사용 가능
 - b. driver, url, driver, password
 - <sql:setDataSource driver="oracle.jdbc.driver.OracleDriver" url="jdbc:oracle:thin:@localhost:1521:XE" user="scott" password="tiger" />
- Connection 정보는 var 속성에 입력된 변수에 저장

[server.xml]

```
<Resource auth="Container" driverClassName="oracle.jdbc.driver.OracleDriver"
maxActive="10" minIdle="10" maxWait="1000" name="jdbc/myoracle"
password="tiger" type="javax.sql.DataSource"
url="jdbc:oracle:thin:@localhost:1521:XE" username="scott"/>
```

[web.xml]

```
<resource-ref>
<description>JNDI Test</description>
<res-ref-name>jdbc/myoracle</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```

[sqldemo.jsp]

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<body>
<c:catch var="error">
<!-- 1st 방법 --%>
<!-- <sql:setDataSource driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/world"
user="root" password="javamysql" var="conn"/> --%>
<!-- 2nd 방법 --%>
<sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
<c:out value="${conn}" />
</c:catch>
<c:if test="${!empty error}">
Error Message : <c:out value="${error}" />
</c:if>
</body>
```

5) <sql:query>

- Database server에 query를 전송할 때 사용
- <sql:query var="변수명" scope="범위" sql="sql" dataSource="dataSource" startRow="startRow" maxRows="maxRows" />
- dataSource : JNDI resource나 <sql:setDataSource>로 얻어온 변수명으로 설정
- sql : 실행할 SQL 문 지정 또는 body에서 설정 가능
- var : sql query가 실행된 결과 저장
- startRow : 얻어진 query 결과의 시작 행 값을 의미, 1번째 record 값부터 시작하려면 0으로 설정
- maxRow : 얻어진 query 결과의 record 최대 수를 의미
- Result interface
 - See javax.servlet.jsp.jstl.sql.Result interface

6) Result interface

- This interface represents the result of a <sql:query> action.
- It provides access to the following information in the query result:
 - The result rows --> getRows() and getRowsByIndex()
 - The column names --> getColumnNames()
 - The number of rows in the result --> getRowCount()
- An indication whether the rows returned represent the complete result or just a subset that is limited by a maximum row setting --> isLimitedByMaxRows()

```

1001 -Methods
1002     --java.lang.String [] getColumnNames()
1003     --int getRowCount()
1004     --java.util.SortedMap[] getRows()
1005     --java.lang.Object [][] getRowsByIndex()
1006     --boolean isLimitedByMaxRows()
1007
1008 [sqldemo1.jsp]
1009 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1010 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1011 <!DOCTYPE html>
1012 <html lang="ko">
1013 <head>
1014     <style>
1015         table {
1016             width:600px;border-collapse:collapse;
1017             margin:auto;
1018         }
1019         th, td{
1020             border:1px solid black;
1021         }
1022         td{
1023             text-align:center;
1024         }
1025         caption{
1026             font-size:1.5em;color:blue;
1027         }
1028     </style>
1029 </head>
1030 <body>
1031 <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"
1032                     url="jdbc:oracle:thin:@localhost:1521:xe"
1033                     user="scott" password="tiger" />
1034 <sql:query var="rs" dataSource="${conn}">
1035     SELECT empno, ename, job, hiredate, sal, comm FROM emp
1036 </sql:query>
1037 <table>
1038     <caption>사원 정보</caption>
1039     <thead>
1040         <tr height="40">
1041             <th width="100">사원번호</th><th width="100">사원명</th><th width="100">업무</th>
1042             <th width="100">입사일</th><th width="100">봉급</th><th width="100">보너스</th>
1043         </tr>
1044     </thead>
1045     <tbody>
1046         <c:forEach var="data" items="${rs.rows}">
1047             <tr>
1048                 <td><c:out value="${data.empno}"/></td>
1049                 <td><c:out value="${data.ename}"/></td>
1050                 <td><c:out value="${data.job}"/></td>
1051                 <td><c:out value="${data.hiredate}"/></td>
1052                 <td><c:out value="${data.sal}"/></td>
1053                 <td><c:out value="${data.comm}"/></td>
1054             </tr>
1055         </c:forEach>
1056     </tbody>
1057 </table>
1058 </body>
1059 </html>
1060
1061 [sqldemo2.jsp]
1062 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1063 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1064 <!DOCTYPE html>
1065 <html lang="ko">
1066 <head>
1067     <style>

```



```

1068     table {
1069         width:600px;border-collapse:collapse;
1070         margin:auto;
1071     }
1072     th, td{
1073         border:1px solid black;width:200px;
1074     }
1075     td{
1076         text-align:center;
1077     }
1078     caption{
1079         font-size:1.5em;color:blue;
1080     }
1081 </style>
1082 </head>
1083 <body>
1084 <sql:query var="rs" dataSource="jdbc/myoracle">
1085     SELECT deptno, dname, loc FROM dept ORDER BY deptno DESC
1086 </sql:query>
1087 <table>
1088     <caption>부서 정보</caption>
1089     <thead>
1090         <tr height="40">
1091             <c:forEach var="columnName" items="{rs.columnNames}">
1092                 <th><c:out value="{columnName}" /></th>
1093             </c:forEach>
1094         </tr>
1095     </thead>
1096     <tbody>
1097         <c:forEach var="row" items="{rs.rowsByIndex}">
1098             <tr height="50">
1099                 <c:forEach var="column" items="{row}" varStatus="i">
1100                     <td>
1101                         <c:if test="{column != null}">
1102                             <c:out value="{column}" />
1103                         </c:if>
1104                         <c:if test="{column == null}">
1105                             &nbsp;
1106                         </c:if>
1107                     </td>
1108                 </c:forEach>
1109             </tr>
1110         </c:forEach>
1111     </tbody>
1112 </table>
1113 </body>
1114 </html>
1115

```

7)<sql:update>

- Record의 추가, 수정, 삭제 기능 담당
- <sql:update var="변수명" scope="범위" sql="sql" dataSource="dataSource" />
- var 속성에 입력한 변수에는 레코드를 업데이트한 결과가 저장
- sql : record를 업데이트할 SQL 문
- dataSource : JNDI resource나 <sql:setDataSource>로 얻어온 변수명으로 설정

[sqldemo3.jsp]

```

1124 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1125 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1126 <!DOCTYPE html>
1127 <html lang="ko">
1128 <body>
1129 <sql:update dataSource="jdbc/myoracle" var="result">
1130     INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')
1131 </sql:update>
1132 <h2>Insert Success</h2>
1133 <c:out value="{result}" />
1134 </body>

```

```

1135     </html>
1136
1137 8)<sql:param>
1138     -<sql:query> tag나 <sql:update> tag로 SQL 문장을 전송할 때 parameter를 전달해 주는 tag
1139     -<sql:param value="value" />
1140     -전달해야 할 parameter가 두 가지 이상일 경우는 순차적으로 사용
1141
1142     [sqldemo4.jsp]
1143     <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1144     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1145     <!DOCTYPE html>
1146     <html lang="ko">
1147     <body>
1148     <sql:update dataSource="jdbc/myoracle" var="result">
1149         INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')
1150         <sql:param value="60" />
1151         <sql:param value="IT" />
1152         <sql:param value="PUSAN" />
1153     </sql:update>
1154     <c:if test="${result == 1}">
1155         <h2>Insert Success</h2>
1156     </c:if>
1157     </body>
1158     </html>
1159
1160     [sqldemo5.jsp]
1161     <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1162     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1163     <!DOCTYPE html>
1164     <html lang="ko">
1165     <head>
1166         <style>
1167             table {
1168                 width:600px;border-collapse:collapse;
1169                 margin:auto;
1170             }
1171             th, td{
1172                 border:1px solid black;width:200px;
1173             }
1174             td{
1175                 text-align:center;
1176             }
1177             caption{
1178                 font-size:1.5em;color:blue;
1179             }
1180         </style>
1181     </head>
1182     <body>
1183     <sql:query var="rs" dataSource="jdbc/mymysql">
1184         SELECT zipcode, sido, gugun, dong, bunji FROM zipcode
1185         WHERE dong LIKE CONCAT('%', ?, '%')
1186         <sql:param value="${param.keyword}" />
1187     </sql:query>
1188     <table>
1189         <caption>우편번호검색</caption>
1190         <thead>
1191             <tr height="40">
1192                 <c:forEach var="columnName" items="${rs.columnNames}">
1193                     <th><c:out value="${columnName}" /></th>
1194                 </c:forEach>
1195             </tr>
1196         </thead>
1197         <tbody>
1198             <c:forEach var="row" items="${rs.rowsByIndex}">
1199                 <tr height="50">
1200                     <c:forEach var="column" items="${row}" varStatus="i">
1201                         <td>

```

```

1202         <c:if test="${column != null}">
1203             <c:out value="${column}" />
1204         </c:if>
1205         <c:if test="${column == null}">
1206             &nbsp;
1207         </c:if>
1208     </td>
1209 </c:forEach>
1210 </tr>
1211 </c:forEach>
1212 </tbody>
1213 </table>
1214 </body>
1215 </html>

```

9)<sql:dateParam>

```

1218 -<sql:param> 과 같이 parameter를 전달할 때 사용
1219 -<sql:dateParam value="value" type="type" />
1220 -차이점은 이 tag는 날짜 정보의 parameter를 전달할 때 사용
1221 -type 속성으로 data, time, timestamp 중 하나의 값 가능
1222 -Statement 의 setDate(), setTime(), setTimestamp() method와 같은 기능

```

10)<sql:transaction >

```

1225 -Database 작업 시 transaction을 사용하도록 하는 tag
1226 -<sql:transaction dataSource="dataSource" isolation="isolation" />
1227 -dataSource : JNDI resource나 <sql:setDataSource>로 얻어온 변수명으로 설정
1228 -isolation 속성을 이용하여 격리 수준을 지정가능
1229 -가능한 값은 read_committed, read_uncommitted, repeatable_read, serializable
1230 -이 tag의 body 부분에 query tag를 사용하며, 실패할 경우는 transaction tag내의 작업을 모두 반영하지 않기
    때문에 transaction 관리가 가능

```

[sqldemo6.jsp]

```

1231 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1232 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1233 <!DOCTYPE html>
1234 <html lang="ko">
1235 <head>
1236     <style>
1237         table {
1238             width:600px;border-collapse:collapse;
1239             margin:auto;
1240         }
1241         th, td{
1242             border:1px solid black;width:200px;
1243         }
1244         td{
1245             text-align:center;
1246         }
1247         caption{
1248             font-size:1.5em;color:blue;
1249         }
1250     </style>
1251 </head>
1252 <body>
1253 <c:catch var="ex">
1254 <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"
1255     url="jdbc:oracle:thin:@localhost:1521:xe"
1256     user="scott" password="tiger" />
1257 <sql:update dataSource="${conn}">
1258     INSERT INTO dept VALUES(?,?,?)
1259     <sql:param value="${param.deptno}" />
1260     <sql:param value="${param.dname}" />
1261     <sql:param value="${param.loc}" />
1262 </sql:update>
1263 <sql:query var="rs" dataSource="${conn}">
1264     SELECT deptno, dname, loc FROM dept
1265 </sql:query>

```

```

1268 <table>
1269   <caption>부서정보</caption>
1270   <thead>
1271     <tr height="40">
1272       <th>부서번호</th> <th>부서이름</th> <th>위치</th>
1273     </tr>
1274   </thead>
1275   <tbody>
1276     <c:forEach var="row" items="${rs.rowsByIndex}">
1277       <tr height="50">
1278         <c:forEach var="column" items="${row}">
1279           <td>
1280             <c:if test="${column != null}">
1281               <c:out value="${column}" />
1282             </c:if>
1283             <c:if test="${column == null}">
1284               &nbsp;
1285             </c:if>
1286           </td>
1287         </c:forEach>
1288       </tr>
1289     </c:forEach>
1290   </tbody>
1291 </table>
1292 </c:catch>
1293 <c:if test="${!empty(ex)}">
1294   ERROR : <c:out value="${ex}" />
1295 </c:if>
1296 </body>
1297 </html>

```

11. Function

- 1) 주로 문자열 처리
- 2) java.lang.String class에 속하는 method와 비슷한 기능
- 3) <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
- 4) substring(str, index1, index2)
 - str의 index1부터 index2 - 1 까지의 부분 문자열을 return
- 5) substringAfter(str1, str2)
 - str1에서 str2를 찾아서 그 후의 부분 문자열을 return
 - `${fn:substringAfter("Stella Cadente", "e")}`
 - > lla Cadente
- 6) substringBefore(str1, str2)
 - str1에서 str2를 찾아서 그 전의 부분 문자열을 return
 - `${fn:substringBefore("Stella Cadente", "e")}`
 - > St
- 7) toUpperCase(str)
- 8) toLowerCase(str)
- 9) trim(str)
- 10) replace(str, src, dest)
 - str 문자열에 포함된 src를 모두 dest로 치환한 결과를 return
 - `${fn:replace("Stella Cadente", "e", "E")}`
- 11) indexOf(str1, str2)
 - str1에 포함된 str2의 시작 index를 return
 - `${fn:indexOf("Stella Cadente", "Cadente")}` --> 7
- 12) startsWith(str1, str2)
 - str1이 str2로 시작하면 true, 아니면 false

```

1335 13)endsWith(str1, str2)
1336     -str1이 str2로 끝나면 true, 아니면 false
1337     ${fn:endsWith("Hello World", "World")}
1338
1339 14)contains(str1, str2)
1340     <c:set var="myString" value="Hello World" />
1341     ${fn:contains(myString, "Hello")}
1342     ${fn:contains("Stella Cadente", "Cadente")}
1343
1344 15)containsIgnoreCase(str1, str2)
1345     ${fn:contains("Stella Cadente", "CADENTE")}
1346
1347 16)split(str1, str2)
1348     -str1을 str2를 기준으로 분리해서 만든 부분자열들의 배열을 return
1349     <c:set var="split" value="${fn:split('my,world', ',')}"/>
1350     <table>
1351         <c:forEach var="substring" items="${split}">
1352             <tr><td>${substring}</td></tr>
1353     </table>
1354
1355     <table>
1356         <tr><td>my</td></tr>
1357         <tr><td>world</td></tr>
1358     </table>
1359
1360 17)join(arr, str2)
1361     -arr배열의 모든 항목을 합쳐서 return.
1362     -항목사이에는 str2가 들어감
1363
1364 18)escapeXml(str)
1365     -HTML 문법에 의해 특수 문자로 취급되는 모든 문자표를 이스케이프 시퀀스로 치환한 결과를 return
1366     ${fn:escapeXml("Use <br /> to change lines")}
1367     --> Use &lt;br /&gt; to change lines
1368
1369 19)length(obj)
1370     -obj 가 문자열이면 문자열의 길이, List 나 Collection이면 항목의 수를 return
1371     ${fn:length("Stella Cadente")} --> 14
1372
1373     [fndemo.jsp]
1374     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1375     <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
1376     <html>
1377     <head><title>함수 사용</title></head>
1378     <body>
1379         <c:set var="str1" value="Functions <태그>를 사용합니다. " />
1380         <c:set var="str2" value="사용" />
1381         <c:set var="tokens" value="1,2,3,4,5,6,7,8,9,10" />
1382
1383         length(str1) = ${fn:length(str1)} <br>
1384         toUpperCase(str1) = "${fn:toUpperCase(str1)}" <br>
1385         toLowerCase(str1) = "${fn:toLowerCase(str1)}" <br>
1386         substring(str1, 3, 6) = "${fn:substring(str1, 3, 6)}" <br>
1387         substringAfter(str1, str2) = "${fn:substringAfter(str1, str2)}" <br>
1388         substringBefore(str1, str2) = "${fn:substringBefore(str1, str2)}" <br>
1389         trim(str1) = "${fn:trim(str1)}" <br>
1390         replace(str1, src, dest) = "${fn:replace(str1, " ", "-")}" <br>
1391         indexOf(str1, str2) = "${fn:indexOf(str1, str2)}" <br>
1392         startsWith(str1, str2) = "${fn:startsWith(str1, 'Fun')}" <br>
1393         endsWith(str1, str2) = "${fn:endsWith(str1, "합니다.")}" <br>
1394         contains(str1, str2) = "${fn:contains(str1, str2)}" <br>
1395         containsIgnoreCase(str1, str2) = "${fn:containsIgnoreCase(str1, str2)}" <br>
1396
1397         <c:set var="array" value="${fn:split(tokens, ',')}"/>
1398
1399         join(array, "-") = "${fn:join(array, '-')}" <br>
1400         escapeXml(str1) = "${fn:escapeXml(str1)}" <br>
1401

```

