

```
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 : 콜렉션이나 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 가
    함을 의미
100         false : 정적 문자열만 할당가능
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
149 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='${userid}' />님!!!";
305             alert(str);
306         </script>
307     </c:if>
308
309     8)<c:remove>
310     -설정된 값 제거하는 tag
311     -<c:remove var="변수명" [scope="범위"] />
312     -var : 설정된 속성이 저장되어 있는 변수
313     -scope : 지정된 범위에서 지정된 속성을 제거
314
315     [removedemo.jsp]
316     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
317     <body>
318     <c:set var="test" value="Hello JSTL!"/>
319     <h3>&lt;c:set&gt; 사용 후 : <c:out value="${test}"/></h3>
320     <c:remove var="test"/>
321     <h3>&lt;c:remove&gt; 사용 후 : <c:out value="${test}"/></h3>
322     </body>
323

```

```

324 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>
335
336     [catchdemo1.jsp]
337     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
338     <c:catch var="ex">
339         name 파라미터의 값 = <%= request.getParameter("name") %><br>
340         <% if (request.getParameter("name").equals("test")) { %>
341             ${param.name}은 test 입니다.
342         <% } %>
343     </c:catch>
344     <p>
345     <c:if test="\${ex != null}">
346         예외가 발생하였습니다:<br>
347         ${ex}
348     </c:if>
349
350 10)<c:if>
351     -조건 처리할 때 사용되는 tag
352     -<c:if test="조건" var="변수명" [scope="범위"] />
353     -test(rtexprvalue = true) : 조건을 지정
354     -var : 속성의 변수에는 조건 처리할 결과를 저장
355     -scope : var 속성에 지정한 변수의 범위를 의미
356
357     [ifdemo.jsp]
358     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
359     <c:if test="true">
360         무조건 수행된다.
361     </c:if>
362     <br />
363
364     <c:if test="\${5<10}">
365     <h3>5는 10보다 작다.</h3>
366     </c:if>
367     <c:if test="\${6+3==9}">
368     <h3>6 + 3 은 9이다.</h3>
369     </c:if>
370     <br />
371
372     <c:if test="\${param.name eq 'code'}" >
373         name 파라미터의 값은 ${param.name} 입니다.
374     </c:if>
375     <br />
376
377     <c:if test="\${!empty param.name}" >

```

```

378     Name : ${param.name}
379 </c:if>
380 <c:if test="${param.age > 18}" >
381     귀하는 성년이시군요.
382 </c:if>
383
384 11)<c:choose>
385     -조건 처리할 때 사용되는 tag
386     -<c:choose>
387         <c:when test="조건"></c:when>
388         <c:otherwise></c:otherwise>
389     </c:choose>
390     -Java의 switch 와 비슷
391     -<c:when> tag의 test 속성으로 조건 확인하며, 조건에 만족하면
392         <c:when> 과 </c:when> 사이에 있는 내용 처리
393     -<c:when> tag는 조건에 따라 여러 개 사용가능
394     -어느 조건도 만족하지 않을 때 <c:otherwise> 실행
395
396 [choosedemo.jsp]
397 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
398 <c:choose>
399     <c:when test="${5+10==50}">
400         <h3>5+10은 50이다.</h3>
401     </c:when>
402
403     <c:otherwise>
404         <h3>5+10은 50이 아니다.</h3>
405     </c:otherwise>
406 </c:choose>
407
408 [choosedemo1.jsp]
409 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
410 <body>
411 <c:choose>
412     <c:when test="${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
478 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
489 [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} = ${j * i}
515                 </td>
516             </c:forEach>
517         </tr>
518     </c:forEach>
519 </table>
520
521 [foreachdemo1.jsp]
522 <%@ page import="java.util.HashMap" %>
523 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
524 <%
525     HashMap mapObj = new HashMap();
526     mapObj.put("name", "조성모");
527     mapObj.put("today", new java.util.Date());
528 %>
529 <c:set var="intArray" value="<%= new int[] {1,2,3,4,5} %>" />
530 <h4>int형 배열</h4>
531
532 <c:forEach var="i" items="${intArray}" begin="2" end="4">
533     [${i}]
534 </c:forEach>
535 <c:set var="map" value="<%=mapObj %>" />
536
537 <h2>Map</h2>
538 <c:forEach var="i" items="${map}">
539     ${i.key} = ${i.value}<br />

```



```

592
593 14)<c:import>
594     -지정된 URL tag가 사용된 JSP page에 출력시키는 기능
595     -<jsp:include> 와 비슷
596     -<c:import url="URL 값" var="변수명" scope="범위" varReader="입력스트림명"
597       context="contextName" charEncoding="인코딩값" />
598     -url : HTTP 뿐만 아니라 FTP 외부 resource도 가능
599     -var : resource가 저장될 변수
600     -scope : var 속성에 지정한 변수의 범위를 의미
601     -varReader : var 속성과 마찬가지로 resource가 저장될 변수를 의미. 단, 이 변수는 Reader 객체이어야 한다.
602     -context : URL에 접근할 때의 context 이름을 의미
603     -charEncoding : 지정된 URL의 resource를 가져올 때 어떻게 encoding할 것인지를 의미
604
605 [importdemo.jsp]
606 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
607 <c:import url="/fortokendsdemo.jsp" var="myurl"></c:import>
608 <body>
609     <h1>Import 연습하기</h1>
610     ${myurl}
611 </body>
612
613 [footer.jsp]
614 <%@ page contentType = "text/html; charset=utf-8" %>
615 <hr size="1">
616 <center>footer area</center>
617
618 [header.jsp]
619 <%@ page contentType = "text/html; charset=utf-8" %>
620 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
621 <hr size="1">
622 <c:if test="${param.id != null}">
623 [HEADER] ${param.id} 님의 방문을 환영합니다.
624 <hr size="1">
625 </c:if>
626
627 [importdemo1.jsp]
628 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
629 <c:import url="setdemo.jsp" var="urlEx" />
630 <c:import url="header.jsp" var="head">
631     <c:param name="id" value="javaexpert" />
632 </c:import>
633 <c:import url="footer.jsp" var="foot" />
634
635 <html>
636 <head><title>import와 url tag</title></head>
637 <body>
638
639     ${head}
640
641     ${urlEx}
642
643     ${foot}
644
645 </body>

```

```

646     </html>
647
648 15)<c:redirect>
649     -지정된 URL로 page를 이동시키는 기능
650     -<c:redirect url="URL 값" context="contextName" />
651     -url : 이동할 URL 입력
652     -context : context 이름을 입력하지만, 일반적으로 url 속성만 사용
653
654     [redirectdemo.jsp]
655     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
656     <c:redirect url="multiply.jsp">
657         <c:param name="num1" value="5" />
658         <c:param name="num2" value="25" />
659     </c:redirect>
660
661 16)<c:url>
662     -value 속성에 지정된 값으로 URL 을 생성하는 기능
663     -<c:url var="변수명" scope="범위" value="값" context="contextName" />
664     -scope : var 속성에 입력되어 있는 변수의 범위를 의미
665
666     [urldemo.jsp]
667     <%@ 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
685 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
691
692 9. I18N capable formatting(국제화)
693 1)국제화/형식화의 기능을 제공해 주는 library
694 2)국제화는 다국어 내용을 처리, 형식화는 날짜와 숫자 형식 등을 처리
695 3)<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
696 4)<fmt:requestEncoding>
697     -request 객체로부터 전달 받은 값들을 encoding할 때 사용
698     -<frm:requestEncoding value="encoding값" />
699     -request.setCharacterEncoding와 같다.

```

```

700
701 [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="${param.irum}"> </c:out>
715 </body>
716 </html>
717
718 5)<fmt:setLocale>
719     -다국어 page를 사용할 때 언어를 지정하는 tag
720     -<frm:setLocale value="값" variant="" scope="범위" />
721     -여러 나라에 해당하는 언어를 지원하도록 설계되어 있을 때 어떤 언어를 사용하여 page를 표시할지 설정하는 tag
722     -value : 어떤 언어를 사용할 지 지정하는 속성이며 언어 code를 입력
723     -variant : Browser의 spec을 지정할 때 사용
724
725 6)<fmt:timeZone>
726     -지정한 지역 값으로 시간대를 맞출 때 사용
727     -<fmt:timeZone value="값" />
728     -<fmt:setTimeZone>과 비슷하지만, <fmt:setTimeZone>은 특정 page 전체에 적용
729     -하지만, 이 tag는 첫 tag와 끝 tag 사이의 body 부분의 내용에만 적용
730     -java.util.TimeZone 값 지정 가능
731
732     <fmt:timeZone value="GMT+1:00">
733         <fmt:formatDate value="${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 차이이다.

```

```

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 -설정된 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
ASCII로 변환
769 [bundle.properties]
770 sendMessage=\uC548\uB155\uD558\uC138\uC694
771 --'안녕하세요'를 한글로 입력하면 자동으로 변환됨
772
773 [frmdemo1.jsp]
774 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
775 <body>
776 <fmt:setBundle basename="bundle" var="myBundle" />
777 출력 메시지 : <fmt:message key="sendMessage" bundle="${myBundle}" />
778
779 11)<fmt:param>
780 -<fmt:message> tag로 읽어 온 resource 내용에 parameter를 전달하는 tag
781 -resource 내용이 만일 '{0}'님 안녕하세요'일 경우 <fmt:param> tag로 {0} 대신 원하는 parameter 지정
782
783 12)<fmt:formatNumber>
784 -숫자 형식의 pattern 설정할 때 사용
785 -<fmt:formatNumber value="값" [type="{number | currency | percent}"]
786 [pattern="패턴"] [currencyCode="값"] [currencySymbol="값"]
787 [groupingUsed="{true | false}"] [maxIntegerDigits="값"] [minIntegerDigits="값"]
788 [maxFractionDigits="값"] [minFractionDigits="값"] [var="변수명"] [scope="범위"] />
789 -value(rtexprvalue = true) : pattern을 적용시킬 원래의 값
790 -type(rtexprvalue = true) : 숫자의 type, type은 number, currency, percent 중 원하는 type 설정
791 -pattern(rtexprvalue = true) : 지정한 값을 어떤 pattern으로 변화시킬지 결정
792 -currencyCode(rtexprvalue = true) : 통화 code를 의미, 숫자 type이 통화일 경우만 유효
793
794 -<ISO 4217의 통화 코드 지정>
795 캐나다 달러 CAD 달러 센트
796 중국 위안 CNY 위안 자오
797 유로 EUR 유로 유로 센트
798 일본 엔 JPY 엔 엔
799 영국 파운드 GBP 파운드 펜스
800 미국 달러 USD 달러 센트
801
802 -currencySymbol(rtexprvalue = true) 도 숫자 type이 통화일때만 유효, 통화 기호 지정
803 -groupingUsed(rtexprvalue = true) : 그룹 기호의 포함 여부
804 -maxIntegerDigits(rtexprvalue = true) : 최대 정수 길이
805 -minIntegerDigits(rtexprvalue = true) : 최소 정수 길이
806 -maxFractionDigits(rtexprvalue = true) : 최대 소수점 자릿수

```



```

807 -minFractionDigits(rtexprvalue = true) : 최소 소수점 자릿수
808
809 ex)
810 <...value="12" type="number" /> --> 12
811 <...value="12" type="number" minIntegerDigits="3" /> --> 012
812 <...value="12" type="number" minFractionDigits="2" /> --> 12.00
813 <...value="123456.78" pattern=".000" /> --> 123456.780
814 <...value="123456.78" pattern="#,#00.0#" /> --> 123, 456.78
815 <...value="12" type="currency" /> --> $12.00
816 <...value="12" type="currency" currencyCode="GBP" /> --> GBP 12.00
817 <...value="0.12" type="percent" /> --> 12%
818 <...value="0.125" type="percent" minFractionDigits="2" /> --> 12.50%
819
820 [frmdemo2.jsp]
821 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
822 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
823 Number : <fmt:formatNumber value="12345678.3456" type="number" /><br />
824 Currency : <fmt:formatNumber value="1500000"
825     type="currency" currencySymbol="₩"/><br />
826 Currency : <fmt:formatNumber value="1500000"
827     type="currency" currencySymbol="$"/><br />
828 Currency : <fmt:formatNumber value="1500000"
829     type="currency" currencySymbol="¥"/><br />
830 Percent : <fmt:formatNumber value="1500000" type="percent" /><br />
831 <c:set var="now" value="<%=new java.util.Date()%>" />
832 Now is ${now}<br />
833 Now is <c:out value="${now}" /><br />
834 Date : <fmt:formatDate value="${now}" type="date" /><br />
835 Time : <fmt:formatDate value="${now}" type="time" /><br />
836 Both : <fmt:formatDate value="${now}" type="both" /><br />
837
838 <fmt:formatNumber value="50000" type="currency"/><br>
839 <fmt:formatNumber value="0.15" type="percent"/><br>
840 <fmt:formatNumber value="500567300" pattern="###,###,###"/><p>
841
842 <jsp:useBean id="date" class="java.util.Date"/>
843 <fmt:formatDate value="${date}" type="date"/><br>
844 <fmt:formatDate value="${date}" type="time"/><br>
845 <fmt:formatDate value="${date}" type="both"/><p>
846
847 <fmt:formatDate value="${date}" type="both" timeStyle="short" dateStyle="short"/><br>
848 <fmt:formatDate value="${date}" type="both" timeStyle="long" dateStyle="long"/><br>
849
850 13)<fmt:parseNumber>
851 -문자열을 숫자, 통화 또는 percent의 형태로 변환할 때 사용하는 tag
852 -<fmt:parseNumber value="값" [type="{number | currency | percent}"]
853   [pattern="패턴"] [parseLocale="값"] [integerOnly="{true | false}"]
854   [var="변수명"] [scope="범위"] />
855 -value 속성에 문자열 값
856 -type 속성에는 숫자, 통화, percent 중 변환할 타입 설정
857 -pattern 속성에는 value 속성의 값을 설정된 type으로 변환하면서 어떤 pattern을 갖게 할 것인지 지정
858 -parseLocale 은 parsing할 기본 pattern 지정
859 -integerOnly 는 지정된 값 중 정수 부분만 해석할지 여부 결정
860

```



```

861 14)<fmt:formatDate>
862 -날짜 형식의 pattern을 설정할 때 사용되는 tag
863 -<fmt:formatDate value="값" [type="{time | date | both}"]
864   [dateStyle="{ default | short | medium | long | full }"]
865   [timeStyle="{ default | short | medium | long | full }"]
866   [pattern="패턴"] [timeZone="값"] [var="변수명"] [scope="범위"] />
867 -value(rtexprvalue = true) 속성에는 날짜 또는 시간을 입력
868 -type(rtexprvalue = true) 속성으로 날짜인지 시간인지 또는 날짜 시간 둘다 포함한 type인지 지정
869 -dateStyle(rtexprvalue = true) 은 날짜 style 지정, type 속성이 date 또는 both 일 때만
870 -timeStyle(rtexprvalue = true) 은 시간 style 지정, type 속성이 time 또는 both 일 때만
871 -timeZone(rtexprvalue = true) 은 날짜와 시간이 표시될 시간대 지정
872
873 ex)
874 -기본format : <fmt:formatDate value="${now}" />
875 -짧은format : <fmt:formatDate value="${now}" dateStyle="short" />
876 -중간format : <fmt:formatDate value="${now}" dateStyle="medium"/>
877 -긴format : <fmt:formatDate value="${now}" dateStyle="long" />
878 -전체format : <fmt:formatDate value="${now}" dateStyle="full" />
879 - 시간 format은 기본적으로 위의 날짜 format과 같음.
880
881 [fmtdemo2.jsp]
882 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
883 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
884
885 <body>
886 <c:set var="now" value="<%=new java.util.Date() %>"/>
887 KOREA Time : <fmt:formatDate value="${now}"
888   type="both" dateStyle="full" timeStyle="full" /><br />
889 UK Time : <fmt:timeZone value="GMT">
890   <fmt:formatDate value="${now}"
891   type="both" dateStyle="full" timeStyle="full" />
892 </fmt:timeZone>
893 </body>
894
895 15)<fmt:parseDate>
896 -문자열을 날짜와 시간의 형태로 변환하는 tag
897 -<fmt:parseDate value="값" [type="{time | date | both}"]
898   [dateStyle="{ default | short | medium | long | full }"]
899   [timeStyle="{ default | short | medium | long | full }"]
900   [pattern="패턴"]
901   [timeZone="값"]
902   [parseLocale="값"] [var="변수명"] [scope="범위"] />
903 -value 속성에 입력된 문자열 값을 type 속성에 지정된 타입으로 날짜와 시간 형태로 변환
904
905 <c:set var="myDate" value="12/12/2005" />
906 <fmt:parseDate var="formattedDate" type="date" dateStyle="short"
907   value="${myDate}" />
908
909
910 10. JSTL sql
911 1)SQL 관련 기능을 제공해 주는 JSTL library
912 2)Database server에 접근, query 전송, transaction 처리 가능
913 3)<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
914 4)<sql:setDataSource />

```

```

915 -Database에 접근
916 -<sql:setDataSource var="variableName" scope="{page | request | session | application}"
917   {dataSource="dataSource" | driver="driver" url="url" user="user"
918     password="password"} />
918 -접근방법 2가지
919   a. dataSource
920     <sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
921     --dataSource 가 JNDI 로 설정되었으면 바로 <sql:query> 사용 가능
922   b. driver, url, driver, password
923     <sql:setDataSource driver="oracle.jdbc.driver.OracleDriver"
924       url="jdbc:oracle:thin:@localhost:1521:XE" user="scott" password="tiger" />
925 -Connection 정보는 var 속성에 입력된 변수에 저장
926
927 [server.xml]
928 <Resource auth="Container" driverClassName="oracle.jdbc.driver.OracleDriver"
929   maxActive="10" minIdle="10" maxWait="1000" name="jdbc/myoracle"
930   password="tiger" type="javax.sql.DataSource"
931   url="jdbc:oracle:thin:@localhost:1521:XE" username="scott"/>
932
933 [web.xml]
934 <resource-ref>
935   <description>JNDI Test</description>
936   <res-ref-name>jdbc/myoracle</res-ref-name>
937   <res-type>javax.sql.DataSource</res-type>
938   <res-auth>Container</res-auth>
939 </resource-ref>
940
941 [sqldemo.jsp]
942 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
943 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
944 <body>
945   <c:catch var="error">
946     <%-- 1st 방법 --%>
947     <%-- <sql:setDataSource driver="com.mysql.jdbc.Driver"
948       url="jdbc:mysql://localhost:3306/world"
949       user="root" password="javamysql" var="conn"/> --%>
950     <%-- 2nd 방법 --%>
951     <sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
952     <c:out value="${conn}" />
953   </c:catch>
954   <c:if test="${!empty error}">
955     Error Message : <c:out value="${error}" />
956   </c:if>
957 </body>
958
959 5)<sql:query>
960 -Database server에 query를 전송할 때 사용
961 -<sql:query var="변수명" scope="범위" sql="sql" dataSource="dataSource"
962   startRow="startRow" maxRows="maxRows" />
963 -dataSource : JNDI resource나 <sql:setDataSource>로 얻어온 변수명으로 설정
964 -sql : 실행할 SQL 문 지정 또는 body에서 설정 가능
965 -var : sql query가 실행된 결과 저장
966 -startRow : 얻어진 query 결과의 시작 행 값을 의미, 1번째 record 값부터 시작하려면 0으로 설정
967 -maxRow : 얻어진 query 결과의 record 최대 수를 의미

```

```

968 -Result interface
969 --See javax.servlet.jsp.jstl.sql.Result interface
970
971 6)Result interface
972 -This interface represents the result of a <sql:query> action.
973 -It provides access to the following information in the query result:
974 --The result rows --> getRows() and getRowsByIndex()
975 --The column names --> getColumnNames()
976 --The number of rows in the result --> getRowCount()
977 -An indication whether the rows returned represent the complete result or just a subset that
    is limited by a maximum row setting --> isLimitedByMaxRows()
978 -Methods
979 --java.lang.String [] getColumnNames()
980 --int getRowCount()
981 --java.util.SortedMap[] getRows()
982 --java.lang.Object [][] getRowsByIndex()
983 --boolean isLimitedByMaxRows()
984
985 [sqldemo1.jsp]
986 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
987 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
988 <!DOCTYPE html>
989 <html lang="ko">
990 <head>
991 <style>
992 table {
993     width:600px;border-collapse:collapse;
994     margin:auto;
995 }
996 th, td{
997     border:1px solid black;
998 }
999 td{
1000     text-align:center;
1001 }
1002 caption{
1003     font-size:1.5em;color:blue;
1004 }
1005 </style>
1006 </head>
1007 <body>
1008 <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"
1009     url="jdbc:oracle:thin:@localhost:1521:xe"
1010     user="scott" password="tiger" />
1011 <sql:query var="rs" dataSource="${conn}">
1012     SELECT empno, ename, job, hiredate, sal, comm FROM emp
1013 </sql:query>
1014 <table>
1015 <caption>사원 정보</caption>
1016 <thead>
1017 <tr height="40">
1018 <th width="100">사원번호</th><th width="100">사원명</th><th width="100">업무
    </th>
1019 <th width="100">입사일</th><th width="100">봉급</th><th width="100">보너스

```

```

1020         </th>
1021     </tr>
1022 </thead>
1023 <tbody>
1024     <c:forEach var="data" items="${rs.rows}">
1025         <tr>
1026             <td><c:out value="${data.empno}"/></td>
1027             <td><c:out value="${data.ename}"/></td>
1028             <td><c:out value="${data.job}"/></td>
1029             <td><c:out value="${data.hiredate}"/></td>
1030             <td><c:out value="${data.sal}"/></td>
1031             <td><c:out value="${data.comm}"/></td>
1032         </tr>
1033     </c:forEach>
1034 </tbody>
1035 </table>
1036 </body>
1037 </html>
1038 [sqldemo2.jsp]
1039 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1040 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1041 <!DOCTYPE html>
1042 <html lang="ko">
1043 <head>
1044     <style>
1045         table {
1046             width:600px;border-collapse:collapse;
1047             margin:auto;
1048         }
1049         th, td{
1050             border:1px solid black;width:200px;
1051         }
1052         td{
1053             text-align:center;
1054         }
1055         caption{
1056             font-size:1.5em;color:blue;
1057         }
1058     </style>
1059 </head>
1060 <body>
1061 <sql:query var="rs" dataSource="jdbc/myoracle">
1062     SELECT deptno, dname, loc FROM dept ORDER BY deptno DESC
1063 </sql:query>
1064 <table>
1065     <caption>부서 정보</caption>
1066     <thead>
1067         <tr height="40">
1068             <c:forEach var="columnName" items="${rs.columnNames}">
1069                 <th><c:out value="${columnName}" /></th>
1070             </c:forEach>
1071         </tr>
1072     </thead>

```

```

1073     <tbody>
1074         <c:forEach var="row" items="${rs.rowsByIndex}">
1075             <tr height="50">
1076                 <c:forEach var="column" items="${row}" varStatus="i">
1077                     <td>
1078                         <c:if test="${column != null}">
1079                             <c:out value="${column}" />
1080                         </c:if>
1081                         <c:if test="${column == null}">
1082                             &nbsp;
1083                         </c:if>
1084                     </td>
1085                 </c:forEach>
1086             </tr>
1087         </c:forEach>
1088     </tbody>
1089 </table>
1090 </body>
1091 </html>
1092
1093 7)<sql:update>
1094     -Record의 추가, 수정, 삭제 기능 담당
1095     -<sql:update var="변수명" scope="범위" sql="sql" dataSource="dataSource" />
1096     -var 속성에 입력한 변수에는 레코드를 업데이트한 결과가 저장
1097     -sql : record를 업데이트할 SQL 문
1098     -dataSource : JNDI resource나 <sql:setDataSource>로 얻어온 변수명으로 설정
1099
1100     [sqldemo3.jsp]
1101     <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1102     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1103     <!DOCTYPE html>
1104     <html lang="ko">
1105     <body>
1106         <sql:update dataSource="jdbc/myoracle" var="result">
1107             INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')
1108         </sql:update>
1109         <h2>Insert Success</h2>
1110         <c:out value="${result}" />
1111     </body>
1112 </html>
1113
1114 8)<sql:param>
1115     -<sql:query> tag나 <sql:update> tag로 SQL 문장을 전송할 때 parameter를 전달해 주는 tag
1116     -<sql:param value="value" />
1117     -전달해야 할 parameter가 두 가지 이상일 경우는 순차적으로 사용
1118
1119     [sqldemo4.jsp]
1120     <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1121     <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1122     <!DOCTYPE html>
1123     <html lang="ko">
1124     <body>
1125         <sql:update dataSource="jdbc/myoracle" var="result">
1126             INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')

```

```

1127     <sql:param value="60" />
1128     <sql:param value="IT" />
1129     <sql:param value="PUSAN" />
1130 </sql:update>
1131 <c:if test="${result == 1}">
1132     <h2>Insert Success</h2>
1133 </c:if>
1134 </body>
1135 </html>
1136
1137 [sqldemo5.jsp]
1138 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1139 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1140 <!DOCTYPE html>
1141 <html lang="ko">
1142 <head>
1143     <style>
1144         table {
1145             width:600px;border-collapse:collapse;
1146             margin:auto;
1147         }
1148         th, td{
1149             border:1px solid black;width:200px;
1150         }
1151         td{
1152             text-align:center;
1153         }
1154         caption{
1155             font-size:1.5em;color:blue;
1156         }
1157     </style>
1158 </head>
1159 <body>
1160 <sql:query var="rs" dataSource="jdbc/mymysql">
1161     SELECT zipcode, sido, gugun, dong, bunji FROM zipcode
1162     WHERE dong LIKE CONCAT('%', ?, '%')
1163     <sql:param value="${param.keyword}" />
1164 </sql:query>
1165 <table>
1166     <caption>우편번호검색</caption>
1167     <thead>
1168         <tr height="40">
1169             <c:forEach var="columnName" items="${rs.columnNames}">
1170                 <th><c:out value="${columnName}" /></th>
1171             </c:forEach>
1172         </tr>
1173     </thead>
1174     <tbody>
1175         <c:forEach var="row" items="${rs.rowsByIndex}">
1176             <tr height="50">
1177                 <c:forEach var="column" items="${row}" varStatus="i">
1178                     <td>
1179                         <c:if test="${column != null}">
1180                             <c:out value="${column}" />

```

```

1181         </c:if>
1182         <c:if test="${column == null}">
1183             &nbsp;
1184         </c:if>
1185     </td>
1186 </c:forEach>
1187 </tr>
1188 </c:forEach>
1189 </tbody>
1190 </table>
1191 </body>
1192 </html>

```

9)<sql:dateParam>

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

10)<sql:transaction >

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

[sqldemo6.jsp]

```

1210 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
1211 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1212 <!DOCTYPE html>
1213 <html lang="ko">
1214 <head>
1215     <style>
1216         table {
1217             width:600px;border-collapse:collapse;
1218             margin:auto;
1219         }
1220         th, td{
1221             border:1px solid black;width:200px;
1222         }
1223         td{
1224             text-align:center;
1225         }
1226         caption{
1227             font-size:1.5em;color:blue;
1228         }
1229     </style>
1230 </head>
1231 <body>
1232 <c:catch var="ex">
1233 <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"

```

```

1234         url="jdbc:oracle:thin:@localhost:1521:xe"
1235         user="scott" password="tiger" />
1236     <sql:update dataSource="${conn}">
1237         INSERT INTO dept VALUES(?,?,?)
1238     <sql:param value="${param.deptno}" />
1239     <sql:param value="${param.dname}" />
1240     <sql:param value="${param.loc}" />
1241 </sql:update>
1242 <sql:query var="rs" dataSource="${conn}">
1243     SELECT deptno, dname, loc FROM dept
1244 </sql:query>
1245 <table>
1246     <caption>부서정보</caption>
1247     <thead>
1248         <tr height="40">
1249             <th>부서번호</th> <th>부서이름</th> <th>위치</th>
1250         </tr>
1251     </thead>
1252     <tbody>
1253         <c:forEach var="row" items="${rs.rowsByIndex}">
1254             <tr height="50">
1255                 <c:forEach var="column" items="${row}">
1256                     <td>
1257                         <c:if test="${column != null}">
1258                             <c:out value="${column}" />
1259                         </c:if>
1260                         <c:if test="${column == null}">
1261                             &nbsp;
1262                         </c:if>
1263                     </td>
1264                 </c:forEach>
1265             </tr>
1266         </c:forEach>
1267     </tbody>
1268 </table>
1269 </c:catch>
1270 <c:if test="${!empty(ex)}">
1271     ERROR : <c:out value="${ex}" />
1272 </c:if>
1273 </body>
1274 </html>

```

1275

1276

1277 11. Function

- 1278 1)주로 문자열 처리
- 1279 2)java.lang.String class에 속하는 method와 비슷한 기능
- 1280 3)<%@ taglib prefix="fn" uri="<http://java.sun.com/jsp/jstl/functions>" %>
- 1281
- 1282 4)substring(str, index1, index2)
- 1283 -str의 index1부터 index2 -1 까지의 부분 문자열을 return
- 1284
- 1285 5)substringAfter(str1, str2)
- 1286 -str1에서 str2를 찾아서 그 후의 부분 문자열을 return
- 1287 \${fn:substringAfter("Stella Cadente", "e")}


```
1288      --> lla Cadente
1289
1290 6)substringBefore(str1, str2)
1291   -str1에서 str2를 찾아서 그 전의 부분 문자열을 return
1292   ${fn:substringBefore("Stella Cadente", "e")}
1293   --> St
1294
1295 7)toUpperCase(str)
1296
1297 8)toLowerCase(str)
1298
1299 9)trim(str)
1300
1301 10)replace(str, src, dest)
1302   -str문자열에 포함된 src를 모두 dest로 치환한 결과를 return
1303   ${fn:replace("Stella Cadente", "e", "E")}
1304
1305 11)indexOf(str1, str2)
1306   -str1에 포함된 str2의 시작 index를 return
1307   ${fn:indexOf("Stella Cadente", "Cadente")} --> 7
1308
1309 12)startsWith(str1, str2)
1310   -str1이 str2로 시작하면 true, 아니면 false
1311
1312 13)endsWith(str1, str2)
1313   -str1이 str2로 끝나면 true, 아니면 false
1314   ${fn:endsWith("Hello World", "World")}
1315
1316 14)contains(str1, str2)
1317   <c:set var="myString" value="Hello World" />
1318   ${fn:contains(myString, "Hello")}
1319   ${fn:contains("Stella Cadente", "Cadente")}
1320
1321 15)containsIgnoreCase(str1, str2)
1322   ${fn:contains("Stella Cadente", "CADENTE")}
1323
1324 16)split(str1, str2)
1325   -str1을 str2를 기준으로 분리해서 만든 부분문자열들의 배열을 return
1326   <c:set var="split" value="${fn:split('my,world', ',')}"/>
1327   <table>
1328     <c:forEach var="substring" items="${split}">
1329       <tr><td>${substring}</td></tr>
1330     </table>
1331
1332   <table>
1333     <tr><td>my</td></tr>
1334     <tr><td>world</td></tr>
1335   </table>
1336
1337 17)join(arr, str2)
1338   -arr배열의 모든 항목을 합쳐서 return.
1339   -항목사이에는 str2가 들어감
1340
1341 18)escapeXml(str)
```

```

1342     -HTML 문법에 의해 특수 문자로 취급되는 모든 문자표를 이스케이프 시퀀스로 치환한 결과를 return
1343     ${fn:escapeXml("Use <br /> to change lines")}
1344     --> Use &lt;br /&gt; to change lines
1345
1346 19)length(obj)
1347     -obj 가 문자열이면 문자열의 길이, List 나 Collection이면 항목의 수를 return
1348     ${fn:length("Stella Cadente")} --> 14
1349
1350 [fndemo.jsp]
1351 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
1352 <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
1353 <html>
1354 <head><title>함수 사용</title></head>
1355 <body>
1356     <c:set var="str1" value="Functions <태그>를 사용합니다. " />
1357     <c:set var="str2" value="사용" />
1358     <c:set var="tokens" value="1,2,3,4,5,6,7,8,9,10" />
1359
1360     length(str1) = ${fn:length(str1)} <br>
1361     toUpperCase(str1) = "${fn:toUpperCase(str1)}" <br>
1362     toLowerCase(str1) = "${fn:toLowerCase(str1)}" <br>
1363     substring(str1, 3, 6) = "${fn:substring(str1, 3, 6)}" <br>
1364     substringAfter(str1, str2) = "${fn:substringAfter(str1, str2)}" <br>
1365     substringBefore(str1, str2) = "${fn:substringBefore(str1, str2)}" <br>
1366     trim(str1) = "${fn:trim(str1)}" <br>
1367     replace(str1, src, dest) = "${fn:replace(str1, " ", "-")}" <br>
1368     indexOf(str1, str2) = "${fn:indexOf(str1, str2)}" <br>
1369     startsWith(str1, str2) = "${fn:startsWith(str1, 'Fun')}" <br>
1370     endsWith(str1, str2) = "${fn:endsWith(str1, "합니다.")}" <br>
1371     contains(str1, str2) = "${fn:contains(str1, str2)}" <br>
1372     containsIgnoreCase(str1, str2) = "${fn:containsIgnoreCase(str1, str2)}" <br>
1373
1374     <c:set var="array" value="${fn:split(tokens, ',')}" />
1375
1376     join(array, "-") = "${fn:join(array, "-")}" <br>
1377     escapeXml(str1) = "${fn:escapeXml(str1)}" <br>
1378
1379 </body>

```