

1. JSTL?
 - 표준화된 태그 라이브러리들을 제공함으로써 보다 편리하게 웹 응용프로그램을 개발할 수 있도록 지원
 - JSTL 1.1 은 Servlet 2.4 이상, JSP 2.0 이상
 - JSTL 1.2 은 Servlet 2.5 이상, JSP 2.1 이상
 - Java Community Process(www.jcp.org)의 JSR-52 가 관리
 2. Homepage
 - 1) JSTL 1.1.x
<http://tomcat.apache.org/taglibs/standard/>
 - 2) JSTL 1.2.x
<https://jstl.java.net/>
 3. Downloads
 - JSTL 을 사용하려면 JSTL API 와 JSTL 구현체를 모두 다운로드해야 함.
 - 1) JSTL 1.1.x
 - <http://archive.apache.org/dist/jakarta/taglibs/standard/binaries/>
 - downloads jakarta-taglibs-standard-1.1.2.zip
 - unzip
 - /lib/jstl.jar, /lib/standard.jar
 - copy to : /WEB-INF/lib
 - 2) JSTL 1.2.x
 - <https://jstl.java.net/download.html>
 - downloads jstl api(javax.servlet.jsp.jstl-api-1.2.1.jar) and jstl implementation(javax.servlet.jsp.jstl-1.2.1.jar)
 - copy to : /WEB-INF/lib
 4. jstl example sites test
 - 1) eclipse에서 standard-doc.war 과 standard-examples.war 파일을 import한다.
import -> WAR files -> standard-examples.war 선택
 - 2) <http://localhost:8080/standard-examples/index.html>
 5. JSTL 에서 제공하는 Tag의 종류
 - 1) <http://jakarta.apache.org/taglibs/doc/standard-doc/standard/GettingStarted.html> 참조하자.
 - 2) Core : <http://java.sun.com/jsp/jstl/core> : c : <c:tagname...>
 - 3) XML processing : <http://java.sun.com/jsp/jstl/xml> : x : <x:tagname...>
 - 4) I18N capable formatting : <http://java.sun.com/jsp/jstl/fmt> : fmt : <fmt:tagname...>
 - 5) Database access(SQL) : <http://java.sun.com/jsp/jstl/sql> : sql : <sql:tagname...>
 - 6) Functions : <http://java.sun.com/jsp/jstl/functions> : fn : fn:functionName(...)
 6. JavaServer Pages Standard Tag Library 1.1 Tag Reference
 - <http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html>
 - >
http://docs.oracle.com/cd/E17802_01/products/products/jsp/jstl/1.1/docs/tlddocs/index.html
 7. JavaServer Pages Standard Tag Library
 - <http://docs.oracle.com/javaee/5/tutorial/doc/bnakc.html>
 - <http://docs.oracle.com/javaee/5/jstl/1.1/docs/api/>
 8. Core Library
 - 1) JSP 페이지에서 필요한 가장 핵심적이고 기본적인 기능들을 제공
 - 2) <%@ taglib prefix="c" uri="<http://java.sun.com/jsp/jstl/core>" %>
- ```
[coredemo.jsp]
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:forEach begin="1" end="10">
 <h3>Hello, World</h3>
</c:forEach>
```
- 3) Tags

```

61 - c:catch : 예외처리에 사용
62 - c:out : JspWriter 에 내용 출력
63 - c:set : JSP 에서 사용될 변수 설정, setAttribute(name, key) 기능과 동일, scope 에 따른 바인딩 처리 가
 능
64 - c:remove : 설정한 변수 제거, removeAttribute(name) 과 동일, scope 에 따른 속성 제거 가능
65 - c:if : 조건처리
66 - c:choose : 다중 조건 처리
67 - c:forEach : 컬렉션이나 Map 의 각 항목 처리
68 - c:forTokens : 구분자로 분리된 각각의 토큰을 처리할 때 사용
69 - c:when
70 - c:otherwise
71 - c:import : URL 을 사용하여 다른 자원의 결과를 삽입
72 - c:redirect : URL 을 재작성한다.
73 - c:param
74
75 *****
76 * 범례 :
77 -required : 필수,
78 -rtexprvalue = true : 정적문자열 및 동적인 값(자바표현식, EL 표현식, <jsp:attribute)로 설정값 모두 가능
 함을 의미
79 false 이면 정적 문자열만 할당가능
80 *****
81
82 4) <c:out>
83 - 지정된 값 출력
84 - <c:out value="출력값" [default="기본값"] [escapeXml="{true | false}"] /> or
85 <c:out value="출력값" [default="기본값"] [escapeXml="{true | false}"]>
86 default value
87 </c:out>
88 - value(rtexprvalue = true, required) 속성을 이용해서 변수의 내용 출력
89 - default(rtexprvalue = true) 속성은 기본 값을 의미, value 값이 null 일 경우 이 기본 값 출력
90 - escapeXml(rtexprvalue = true) 은 false가 기본으로 설정, <, > 등의 특수 기호를 처리할 때 사용,
91 true 이면 < 값은 < 으로 표현, > 값은 > 로 표현
92
93 [outdemo.jsp]
94 <%@ page import = "java.io.IOException, java.io.FileReader" %>
95 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
96 <html>
97 <head><title>소스 보기</title></head>
98 <body>
99 <c:catch var="err">
100 <%
101 FileReader reader = null;
102 String path = request.getParameter("path");
103 reader = new FileReader(getServletContext().getRealPath(path));
104 %>
105 <pre>
106 소스 코드 = <%= path %>
107 <c:out value="<%= reader %>" escapeXml="true" />
108 </pre>
109 </c:catch>
110 <h3>Message : <c:out value="$ {err}"/></h3>
111 </body>
112 </html>
113 http://localhost /outdemo.jsp?path=setdemo.jsp
114
115 [outdemo1.jsp]
116 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
117 <c:out value="커다란 글씨는 다음과 같이 출력합니다" />

118 <c:out value="커다란 글씨는 다음과 같이 출력합니다" escapeXml = "false" />
119

```

```

120 [outdemo2.jsp]
121 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
122 안녕하세요. <c:out value="${param.id}" default="guest" />님!
123
124 5) <c:set>
125 - 지정된 변수에 값을 설정
126 - <c:set var="변수명" value="설정값" target="객체" property="값" [scope="범위"] />
127 - var 는 값을 저장할 변수의 이름
128 - value(rtexprvalue = true) 는 저장할 값
129 - target(rtexprvalue = true) 은 값을 설정할 프라퍼티에 대한 객체(bean객체, Map객체)를 의미
130 - property(rtexprvalue = true) 는 값을 설정할 객체의 프로퍼티
131 - scope 변수의 유효범위, 설정하지 않으면 기본값으로 page
132
133 [setdemo.jsp]
134 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
135 <body>
136 <c:set var="browser" value="${header['user-agent']}" />
137 삭제전

138 <c:out value="${browser}" />

139 <c:remove var="browser"/>
140 삭제후

141 <c:out value="${browser}" />
142 </body>
143
144 [setdemo1.jsp]
145 <%@ page import="java.util.Date" %>
146 <%@ page import="java.util.HashMap" %>
147 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
148 <c:set var="num1" value="${20}" />
149 <c:set var="num2" >
150 10.5
151 </c:set>
152 변수 num1 : ${num1}

153 변수 num2 : ${num2}

154 num1 + num2 = ${num1 + num2}

155
156 <c:set var="today" value="<%=new Date() %>" />
157 Today is ${today}.
158

159 <c:set var="map" value="<%=new HashMap() %>" />
160 <c:set target="${map}" property="irum" value="조성모" />
161 귀하의 이름은 ${map.irum} 이군요.
162
163 [LoginBean.java]
164 package kr.co.javaexpert.libs;
165
166 public class LoginBean {
167 private String userid;
168 private String passwd;
169 public String getUserid() {
170 return userid;
171 }
172 public void setUserid(String userid) {
173 this.userid = userid;
174 }
175 public String getPasswd() {
176 return passwd;
177 }
178 public void setPasswd(String passwd) {
179 this.passwd = passwd;
180 }

```

```
181 }
182
183 [setdemo2.jsp]
184 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
185 <jsp:useBean id="myBean" class="kr.co.javaexpert.libs.LoginBean" />
186
187 <c:set target="${myBean}" property="userid" value="example" />
188 ${myBean.userid}

189 <c:out value="${myBean.userid}" />
190
191 [setdemo3.jsp]
192 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
193 <c:set var="CODE" value="80012" scope="request" />
194 <c:set var="NAME" value="Pencil" scope="request" />
195 <c:set var="PRICE" value="1000" scope="request" />
196 <jsp:forward page="productview.jsp" />
197
198 [productview.jsp]
199 <h3>Product Information</h3>
200 상품코드 : ${CODE}

201 상품명 : ${NAME}

202 단가 : ${PRICE}원
203
204 [com.example.LoginBean.java]
205 public class LoginBean {
206 private String userid;
207 private boolean verified;
208 public String getUserid() {
209 return userid;
210 }
211 public void setUserid(String userid) {
212 this.userid = userid;
213 }
214 public boolean isVerified() {
215 return verified;
216 }
217 public void setVerified(boolean verified) {
218 this.verified = verified;
219 }
220 }
221
222 [login.html]
223 <!DOCTYPE html>
224 <html>
225 <head>
226 <meta charset="UTF-8">
227 <title>로그인해 주세요.</title>
228 </head>
229 <body>
230 <form action="login.jsp" method="post">
231 ID : <input type="text" name="userid" />

232 PWD : <input type="password" name="userpwd" />

233 <input type="submit" value="로그인" />
234 </form>
235 </body>
236 </html>
237
238 [login.jsp]
239 <%@ page language="java" contentType="text/html; charset=UTF-8"
240 pageEncoding="UTF-8"%>
241 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```

242
243 <jsp:useBean id="login" class="com.example.LoginBean" />
244
245 <c:if test="${param.userid eq 'javaexpert' and param.userpwd eq '12345678'}">
246 <jsp:setProperty property="userid" name="login" value="${param.userid}"/>
247 <jsp:setProperty property="verified" name="login" value="true"/>
248 <c:set var="userid" value="${param.userid}" scope="session" />
249 <c:set var="verified" value="true" scope="session" />
250 <c:out value="<div style='color:green'>Login Success</div>"
251 escapeXml="false" />
252 회원전용페이지로...
253 </c:if>
254 <c:if test="${param.userid ne 'javaexpert' or param.userpwd ne '12345678'}">
255 <script>
256 alert("아이디 혹은 비밀번호가 일치하지 않습니다.");
257 history.back();
258 </script>
259 </c:if>
260
261 [secure.jsp]
262 <%@ page language="java" contentType="text/html; charset=UTF-8"
263 pageEncoding="UTF-8"%>
264 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
265
266 <c:set var="verified" value="${sessionScope.verified}" />
267 <c:set var="userid" value="${sessionScope.userid}" />
268
269 <c:if test="${verified ne 'true'}">
270 <script>
271 alert("먼저 로그인부터 해주세요.");
272 location.href = 'login.html';
273 </script>
274 </c:if>
275
276 <c:if test="${verified eq 'true'}">
277 <script>
278 var str = "Welcome!!!<c:out value='${userid}' />님!!!";
279 alert(str);
280 </script>
281 </c:if>
282
283 6) <c:remove>
284 - 설정된 값 제거하는 태그
285 - <c:remove var="변수명" [scope="범위"] />
286 - var 은 설정된 속성이 저장되어 있는 변수
287 - scope 는 지정된 범위에서 지정된 속성을 제거
288
289 [removedemo.jsp]
290 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
291 <body>
292 <c:set var="test" value="Hello JSTL!"/>
293 <h3><c:set> 사용 후 : <c:out value="${test}"/></h3>
294 <c:remove var="test"/>
295 <h3><c:remove> 사용 후 : <c:out value="${test}"/></h3>
296 </body>
297
298 7) <c:catch>
299 - 예외 처리를 위한 태그
300 - <c:catch var="변수명" />
301 - 예외가 발생하면 var(rtexprvalue = true) 에 지정된 예외 객체가 할당
302

```

```

303 [catchdemo.jsp]
304 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
305 <c:catch var="err">
306 <%=10/0%>
307 </c:catch>
308 <h3>Message : <c:out value="${err}"/></h3>
309
310 [catchdemo1.jsp]
311 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
312 <c:catch var="ex">
313 name 파라미터의 값 = <%= request.getParameter("name") %>

314 <% if (request.getParameter("name").equals("test")) { %>
315 ${param.name}은 test 입니다.
316 <% } %>
317 </c:catch>
318 <p>
319 <c:if test="${ex != null}">
320 예외가 발생하였습니다:

321 ${ex}
322 </c:if>
323
324 8) <c:if>
325 - 조건 처리할 때 사용되는 태그
326 - <c:if test="조건" var="변수명" [scope="범위"] />
327 - test(rtexprvalue = true) 는 조건을 지정
328 - var 속성의 변수에는 조건 처리할 결과를 저장
329 - scope 은 var 속성에 지정한 변수의 범위를 의미
330
331 [ifdemo.jsp]
332 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
333 <c:if test="true">
334 무조건 수행된다.
335 </c:if>
336

337
338 <c:if test="${5<10}">
339 <h3>5는 10보다 작다.</h3>
340 </c:if>
341 <c:if test="${6+3==9}">
342 <h3>6 + 3 은 9이다.</h3>
343 </c:if>
344

345
346 <c:if test="${param.name eq 'code'}" >
347 name 파라미터의 값은 ${param.name} 입니다.
348 </c:if>
349

350
351 <c:if test="${!empty param.name}" >
352 Name : ${param.name}
353 </c:if>
354 <c:if test="${param.age > 18}" >
355 귀하는 성년이시군요.
356 </c:if>
357
358 9) <c:choose>
359 - 조건 처리할 때 사용되는 태그
360 - <c:choose>
361 <c:when test="조건"></c:when>
362 <c:otherwise></c:otherwise>
363 </c:choose>

```

```
364 - Java 의 switch 와 비슷
365 - <c:when> 태그의 test 속성으로 조건 확인하며, 조건에 만족하면
366 <c:when> 과 </c:when> 사이에 있는 내용 처리
367 - <c:when> 태그는 조건에 따라 여러 개가 사용가능
368 - 어느 조건도 만족하지 않을 때 <c:otherwise> 실행
369
370 [choosedemo.jsp]
371 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
372 <c:choose>
373 <c:when test="${5+10==50}">
374 <h3>5+10은 50이다.</h3>
375 </c:when>
376
377 <c:otherwise>
378 <h3>5+10은 50이 아니다.</h3>
379 </c:otherwise>
380 </c:choose>
381
382 [choosedemo1.jsp]
383 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
384 <body>
385 <c:choose>
386 <c:when test="${param.name == 'code'}">
387 당신의 이름은 ${param.name} 입니다.

388 </c:when>
389 <c:when test="${param.age > 18}">
390 당신은 미성년자가 아닙니다.
391 </c:when>
392 <c:otherwise>
393 당신은 code도 아니고 미성년자입니다.
394 </c:otherwise>
395 </c:choose>
396 </body>
397
398 [season.html]
399 <!DOCTYPE html>
400 <html>
401 <head>
402 <meta charset="UTF-8">
403 <title>Insert title here</title>
404 </head>
405 <body>
406 <form action="choosedemo.jsp" method="post">
407 당신이 좋아하시는 계절은 ?
408 <input type="text" name="season">
409 <input type="submit" value="전송" />
410 </form>
411 </body>
412 </html>
413
414 [choosedemo.jsp]
415 <%@ page language="java" contentType="text/html; charset=UTF-8"
416 pageEncoding="UTF-8"%>
417 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
418
419 <c:set var="myseason" value="${param.season}" />
420
421 <%-- <c:if test="${myseason eq 'spring'}">
422 <h2>개나리, 진달래</h2>
423 </c:if>
424 <c:if test="${myseason eq 'summer'}">
```

```

425 <h2>장미, 아카시아</h2>
426 </c:if>
427 <c:if test="${myseason eq 'fall'}">
428 <h2>백합, 코스모스</h2>
429 </c:if>
430 <c:if test="${myseason eq 'winter'}">
431 <h2>동백, 매화</h2>
432 </c:if> --%>
433
434 <c:choose>
435 <c:when test="${myseason eq 'spring'}">
436 <h2>개나리, 진달래</h2>
437 </c:when>
438 <c:when test="${myseason eq 'summer'}">
439 <h2>장미, 아카시아</h2>
440 </c:when>
441 <c:when test="${myseason eq 'fall'}">
442 <h2>백합, 코스모스</h2>
443 </c:when>
444 <c:when test="${myseason eq 'winter'}">
445 <h2>동백, 매화</h2>
446 </c:when>
447 <c:otherwise>
448 <h2 style='color:red'>잘못된 계절을 입력하셨습니다.</h2>
449 </c:otherwise>
450 </c:choose>
451
452 10) <c:forEach>
453 - 반복 처리할 때 사용되는 태그, Java 의 for 문과 유사
454 - <c:forEach begin="시작 인덱스" end="끝 인덱스" step="증감식" [var="변수명"] /> or
455 <c:forEach items="객체명" [begin="시작 인덱스"] [end="끝 인덱스"]
456 [step="증감식"] [var="변수명"] [varStatus="상태변수"] />
457 - items(rtexprvalue = true) 속성에 컬렉션이나 배열 형태의 객체를 지정하여 객체의 인덱스만큼 반복 가능
458 - begin(rtexprvalue = true) 과 end(rtexprvalue = true) 속성으로 원하는 범위만큼 반복 수행도 가능
459 - step(rtexprvalue = true) 은 증감식 설정
460 - var는 반복 중일 때 현재 반복하는 있는 값을 기억하는 변수
461 - varStatus는 반복의 상태를 갖는 변수
462
463 [foreachdemo.jsp]
464 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
465 <c:forEach var="test" begin="1" end="10" step="2">
466 ${test}
467 </c:forEach>
468

469
470 <h2>1부터 100까지의 홀수의 합</h2>
471 <c:set var="sum" value="0" />
472 <c:forEach var="i" begin="1" end="100" step="2">
473 <c:set var="sum" value="${sum + i}" />
474 </c:forEach>
475 결과 : ${sum}

476
477 <h2>구구단</h2>
478 <table cellpadding="0" cellspacing="0" border="1" width="800">
479 <tr height="30">
480 <c:forEach var="j" begin="2" end="9">
481 <th width="100">${j}</th>
482 </c:forEach>
483 </tr>
484 <c:forEach var="i" begin="1" end="9">
485 <tr height="20">

```



```

486 <c:forEach var="j" begin="2" end="9">
487 <td width="100">
488 ${j} x ${i} = ${j * i}
489 </td>
490 </c:forEach>
491 </tr>
492 </c:forEach>
493 </table>
494
495 [foreachdemo1.jsp]
496 <%@ page import="java.util.HashMap" %>
497 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
498 <%
499 HashMap mapObj = new HashMap();
500 mapObj.put("name", "조성모");
501 mapObj.put("today", new java.util.Date());
502 %>
503 <c:set var="intArray" value="<%= new int[] {1,2,3,4,5} %>" />
504 <h4>int형 배열</h4>
505
506 <c:forEach var="i" items="${intArray}" begin="2" end="4">
507 [${i}]
508 </c:forEach>
509 <c:set var="map" value="<%=mapObj %>" />
510
511 <h2>Map</h2>
512 <c:forEach var="i" items="${map}">
513 ${i.key} = ${i.value}

514 </c:forEach>
515
516 [foreachdemo2.jsp]
517 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
518 <body>
519 <h1>Header 정보</h1>
520 <c:forEach var="head" items="${headerValues}">
521 param : <c:out value="${head.key}" />

522 value :
523 <c:forEach var="value" items="${head.value}">
524 <c:out value="${value}" />
525 </c:forEach>

526 </c:forEach>
527 </body>
528
529 [foreachdemo3.jsp]
530 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
531 <c:forEach var="cnt" begin="1" end="7">
532 Hello, World

533 </c:forEach>
534
535 11) <c:forTokens>
536 - Java 의 for 문과 StringTokenizer 객체를 결합한 것 같음.
537 - <c:forTokens items="객체명" delims="구분자" [begin="시작 인덱스"] [end="끝 인덱스"]
538 [step="증감식"] [var="변수명"] [varStatus="상태변수"] />
539 - items(rtxprvalue = true) 에 지정한 값을 delims(rtxprvalue = true) 속성의 구분자로 나눈 후
540 나뉜 만큼 반복을 수행
541 - items, delims, var 이 세가지의 속성만으로도 <c:forTokens> 태그 수행 가능
542 - begin(rtxprvalue = true)과 end(rtxprvalue = true)의 경우는 delims 구분자로 나뉜 값들을
543 기준으로 시작 값과 끝 값을 정해주기에 가능
544
545 [fortokensdemo.jsp]
546 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

```



```

608 <c:param name="id" value="javaexpert" />
609 </c:import>
610 <c:import url="footer.jsp" var="foot" />
611
612 <html>
613 <head><title>import와 url 태그</title></head>
614 <body>
615
616 ${head}
617
618 ${urlEx}
619
620 ${foot}
621
622 </body>
623 </html>
624
625 13) <c:redirect>
626 - 지정된 URL 로 페이지를 이동시키는 기능
627 - <c:redirect url="URL 값" context="contextName" />
628 - url 속성에 이동할 URL 입력
629 - context 속성에는 컨텍스트 이름을 입력하지만, 일반적으로 url 속성만 사용
630
631 [redirectdemo.jsp]
632 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
633 <c:redirect url="multiply.jsp">
634 <c:param name="num1" value="5" />
635 <c:param name="num2" value="25" />
636 </c:redirect>
637
638 14) <c:url>
639 - value 속성에 지정된 값으로 URL 을 생성하는 기능
640 - <c:url var="변수명" scope="범위" value="값" context="contextName" />
641 - scope 는 var 속성에 입력되어 있는 변수의 범위를 의미
642
643 [urldemo.jsp]
644 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
645 <c:url value="multiply.jsp" var="abc">
646 <c:param name="num1" value="19" />
647 <c:param name="num2" value="25" />
648 </c:url>
649 <c:redirect url="${abc}" />
650
651 [multiply.jsp]
652 <%@ page language="java" contentType="text/html; charset=UTF-8"
653 pageEncoding="UTF-8"%>
654 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
655
656 <c:set var="first" value="${param.first}" />
657 <c:set var="second" value="${param.second}" />
658
659 <h1>multiply.jsp</h1>
660 ${first} x ${second} = ${first * second}
661
662 15) <c:param>
663 - <c:import> 태그에 파라미터를 전달하기 위한 태그
664 - <c:param name="파라미터명" value="값" />
665 - name 속성에 파라미터명을 입력
666 - value 에 값을 입력하여 전달
667
668

```

## 669 9. I18N capable formatting

- 670 1) 국제화/형식화의 기능을 제공해 주는 라이브러리  
 671 2) 국제화는 다국어 내용을 처리, 형식화는 날짜와 숫자 형식 등을 처리  
 672 3) <%@ taglib prefix="fmt" uri="<http://java.sun.com/jsp/jstl/fmt>" %>  
 673 4) <fmt:requestEncoding>  
 674 - request 객체로부터 전달 받은 값들을 인코딩할 때 사용  
 675 - <frm:requestEncoding value="인코딩값" />  
 676 - request.setCharacterEncoding 와 같다.

```

677
678 [fmtdemo.jsp]
679 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
680 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
681 <fmt:requestEncoding value="utf-8" />
682 <%--
683 <% request.setCharacterEncoding("utf-8"); %>
684 --%>
685
686 <body>
687 <form method="post">
688 Name : <input type="text" name="irum"/>
689 <input type="submit" value="전송" />
690 </form>

691 Name : <c:out value="${param.irum}" /></c:out>
692 </body>
693 </html>
694
695 5) <fmt:setLocale>
696 - 다국어 페이지를 사용할 때 언어를 지정하는 태그
697 - <frm:setLocale value="값" variant="" scope="범위" />
698 - 여러 나라에 해당하는 언어를 지원하도록 설계되어 있을 때 어떤 언어를 사용하여 페이지를 표시할지 설정하
 는 태그
699 - value 속성은 어떤 언어를 사용할 지 지정하는 속성이며 언어 코드를 입력
700 - variant 속성은 브라우저의 스펙을 지정할 때 사용
701
702 6) <fmt:timeZone>
703 - 지정한 지역 값으로 시간대를 맞출 때 사용
704 - <fmt:timeZone value="값" />
705 - <fmt:setTimeZone>과 비슷하지만, <fmt:setTimeZone>은 특정 페이지 전체에 적용
706 - 하지만, 이 태그는 첫 태그와 끝 태그 사이의 body 부분의 내용에만 적용
707 - java.util.TimeZone 값 지정 가능
708
709 <fmt:timeZone value="GMT+1:00">
710 <fmt:formatDate value="${now}" type="both" dateStyle="full"
711 timeStyle="full" />
712 </fmt:timeZone>
713
714 7) <fmt:setTimeZone>
715 - 위와 비슷
716 - <fmt:setTimeZone value="값" [var="변수명"] [scope="범위"] />
717 - var 속성의 변수에 지정한 시간대 값이 저장
718 - scope 로 범위 지정
719
720 8) <fmt:bundle>
721 - properties 확장명을 가진 파일의 리소스를 불러올 때 사용
722 - <fmt:bundle basename="basename" prefix="prefix" />
723 - 보통 다국어 처리 때 사용
724 - basename 속성은 properties 확장명을 가진 파일을 지정
725 - 패키지까지 포함한 경로를 지정해야
726 - prefix는 properties 내의 키 값에 쉽게 접근할 수 있도록 간략한 접근어를 사용
727 - 불러온 리소스는 이 태그의 body 내에서만 사용가능
728

```

```

729 9) <fmt:setBundle>
730 - 위와 같은 기능, 하지만, 사용영역의 차이다.
731 - <fmt:setBundle basename="basename" var="변수명" scope="범위" />
732 - <fmt:bundle> 태그는 첫 태그와 끝 태그 사이의 body 영역에서만 불러온 리소스를 사용하지만,
733 - 이 태그는 페이지 전체에서 사용
734 - var 속성은 설정한 리소스 내용을 가지고 있을 변수명
735 - scope 는 범위를 지정
736
737 10) <fmt:message>
738 - 설정한 properties 파일의 리소스 내용을 읽을 때 사용
739 - <fmt:message key="키값" bundle="bundle변수" var="변수명" scope="범위" />
740 - properties 파일에는 여러 키 값들로 내용이 구분된다.
741 - key 속성값으로 설정된 내용을 가져올 수 있다.
742 - bundle 속성은 <fmt:setBundle>의 var 에서 지정했던 변수를 입력
743 - 이 태그는 bundle 속성을 사용하여 properties 파일에 접근하는 것임.
744
745 /WEB-INF/classes 폴더에 생성, EE Eclipse 를 사용할 경우 src 폴더에 저장하고 한글로 입력하면 자동으로 아
스키로 변환
746 [bundle.properties]
747 sendMessage=\uC548\uB155\uD558\uC138\uC694
748 --'안녕하세요'를 한글로 입력하면 자동으로 변환됨
749
750 [frmdemo1.jsp]
751 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
752 <body>
753 <fmt:setBundle basename="bundle" var="myBundle" />
754 출력 메시지 : <fmt:message key="sendMessage" bundle="${myBundle}" />
755
756 11) <fmt:param>
757 - <fmt:message> 태그로 읽어 온 리소스 내용에 파라미터를 전달하는 태그
758 - 리소스 내용이 만일 '{0}님 안녕하세요'일 경우 <fmt:param> 태그로 {0} 대신 원하는 파라미터 지정
759
760 12) <fmt:formatNumber>
761 - 숫자 형식의 패턴 설정할 때 사용
762 - <fmt:formatNumber value="값" [type="{number | currency | percent}"
763 [pattern="패턴"] [currencyCode="값"] [currencySymbol="값"]
764 [groupingUsed="{true | false}"] [maxIntegerDigits="값"] [minIntegerDigits="값"]
765 [maxFractionDigits="값"] [minFractionDigits="값"] [var="변수명"] [scope="범위"] />
766 - value(rtexprvalue = true) 속성은 패턴을 적용시킬 원래의 값
767 - type(rtexprvalue = true) 은 숫자의 타입, 타입은 number, currency, percent 중 원하는 타입 설정
768 - pattern(rtexprvalue = true) 속성은 지정한 값을 어떤 패턴으로 변화시킬지 결정
769 - currencyCode(rtexprvalue = true) 는 통화 코드를 의미, 숫자 타입이 통화일 경우만 유효
770 <ISO 4217의 통화 코드 지정>
771 캐나다 달러 CAD 달러 센트
772 중국 위안 CNY 위안 자오
773 유로 EUR 유로 유로 센트
774 일본 엔 JPY 엔 센
775 영국 파운드 GBP 파운드 펜스
776 미국 달러 USD 달러 센트
777 - currencySymbol(rtexprvalue = true) 도 숫자 타입이 통화일때만 유효, 통화 기호 지정
778 - groupingUsed(rtexprvalue = true) 는 그룹 기호의 포함 여부
779 - maxIntegerDigits(rtexprvalue = true) 는 최대 정수 길이
780 - minIntegerDigits(rtexprvalue = true) 는 최소 정수 길이
781 - maxFractionDigits(rtexprvalue = true) 는 최대 소수점 자릿수
782 - minFractionDigits(rtexprvalue = true) 는 최소 소수점 자릿수
783
784 ex)
785 <...value="12" type="number" /> --> 12
786 <...value="12" type="number" minIntegerDigits="3" /> --> 012
787 <...value="12" type="number" minFractionDigits="2" /> --> 12.00
788 <...value="123456.78" pattern=".000" /> --> 123456.780

```

```

789 <...value="123456.78" pattern="#,#00.0#" /> --> 123, 456.78
790 <...value="12" type="currency" /> --> $12.00
791 <...value="12" type="currency" currencyCode="GBP" /> --> GBP 12.00
792 <...value="0.12" type="percent" /> --> 12%
793 <...value="0.125" type="percent" minFractionDigits="2" /> --> 12.50%
794
795 [frmdemo2.jsp]
796 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
797 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
798 Number : <fmt:formatNumber value="12345678.3456" type="number" />

799 Currency : <fmt:formatNumber value="1500000"
800 type="currency" currencySymbol="₩"/>

801 Currency : <fmt:formatNumber value="1500000"
802 type="currency" currencySymbol="$"/>

803 Currency : <fmt:formatNumber value="1500000"
804 type="currency" currencySymbol="¥"/>

805 Percent : <fmt:formatNumber value="1500000" type="percent" />

806 <c:set var="now" value="<%=new java.util.Date()%>" />
807 Now is ${now}

808 Now is <c:out value="${now}" />

809 Date : <fmt:formatDate value="${now}" type="date" />

810 Time : <fmt:formatDate value="${now}" type="time" />

811 Both : <fmt:formatDate value="${now}" type="both" />

812
813 <fmt:formatNumber value="50000" type="currency"/>

814 <fmt:formatNumber value="0.15" type="percent"/>

815 <fmt:formatNumber value="500567300" pattern="###,###,###"/><p>
816
817 <jsp:useBean id="date" class="java.util.Date"/>
818 <fmt:formatDate value="${date}" type="date"/>

819 <fmt:formatDate value="${date}" type="time"/>

820 <fmt:formatDate value="${date}" type="both"/><p>
821
822 <fmt:formatDate value="${date}" type="both" timeStyle="short" dateStyle="short"/>

823 <fmt:formatDate value="${date}" type="both" timeStyle="long" dateStyle="long"/>

824
825 13) <fmt:parseNumber>
826 - 문자열을 숫자, 통화 또는 퍼센트의 형태로 변환할 때 사용하는 태그
827 - <fmt:parseNumber value="값" [type="{number | currency | percent}"]
828 [pattern="패턴"] [parseLocale="값"] [integerOnly="{true | false}"]
829 [var="변수명"] [scope="범위"] />
830 - value 속성에 문자열 값
831 - type 속성에는 숫자, 통화, 퍼센트 중 변환할 타입 설정
832 - pattern 속성에는 value 속성의 값을 설정된 타입으로 변환하면서 어떤 패턴을 갖게 할 것인지 지정
833 - parseLocale 은 파싱할 기본 패턴 지정
834 - integerOnly 는 지정된 값 중 정수 부분만 해석할지 여부 결정
835
836 14) <fmt:formatDate>
837 - 날짜 형식의 패턴을 설정할 때 사용되는 태그
838 - <fmt:formatDate value="값" [type="{time | date | both}"]
839 [dateStyle="{ default | short | medium | long | full }"]
840 [timeStyle="{ default | short | medium | long | full }"]
841 [pattern="패턴"] [timeZone="값"] [var="변수명"] [scope="범위"] />
842 - value(rtexprvalue = true) 속성에는 날짜 또는 시간을 입력
843 - type(rtexprvalue = true) 속성으로 날짜인지 시간인지 또는 날짜 시간 둘다 포함한 타입인지 지정
844 - dateStylw(rtexprvalue = true) 은 날짜 스타일 지정, type 속성이 date 또는 both 일 때만
845 - timeStyle(rtexprvalue = true) 은 시간 스타일 지정, type 속성이 time 또는 both 일 때만
846 - timeZone(rtexprvalue = true) 은 날짜와 시간이 표시될 시간대 지정
847
848 ex)
849 - 기본포맷 : <fmt:formatDate value="${now}" />

```

```
850 - 짧은포맷 : <fmt:formatDate value="${now}" dateStyle="short" />
851 - 중간포맷 : <fmt:formatDate value="${now}" dateStyle="medium"/>
852 - 긴포맷 : <fmt:formatDate value="${now}" dateStyle="long" />
853 - 전체포맷 : <fmt:formatDate value="${now}" dateStyle="full" />
854 - 시간 포맷은 기본적으로 위의 날짜 포맷과 같음.
855
856 [fmtdemo2.jsp]
857 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
858 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
859
860 <body>
861 <c:set var="now" value="<%=new java.util.Date() %>"/>
862 KOREA Time : <fmt:formatDate value="${now}"
863 type="both" dateStyle="full" timeStyle="full" />

864 UK Time : <fmt:timeZone value="GMT">
865 <fmt:formatDate value="${now}"
866 type="both" dateStyle="full" timeStyle="full" />
867 </fmt:timeZone>
868 </body>
869
870 15) <fmt:parseDate>
871 - 문자열을 날짜와 시간의 형태로 변환하는 태그
872 - <fmt:parseDate value="값" [type="{time | date | both}"]
873 [dateStyle="{ default | short | medium | long | full }"]
874 [timeStyle="{ default | short | medium | long | full }"]
875 [pattern="패턴"]
876 [timeZone="값"]
877 [parseLocale="값"] [var="변수명"] [scope="범위"] />
878 - value 속성에 입력된 문자열 값을 type 속성에 지정된 타입으로 날짜와 시간 형태로 변환
879
880 <c:set var="myDate" value="12/12/2005" />
881 <fmt:parseDate var="formattedDate" type="date" dateStyle="short"
882 value="${myDate}" />
```