

10장. JSTL(JSP 표준태그 라이브러리)

JSTL



JSTL(JSP 표준태그 라이브러리)

JSTL(Jsp Standart Tag Library)

- Jsp에서 유용하게 사용될 수 있는 기능을 만들어 둔 커스텀 태그 라이브러리이다.
- 핵심(Core), XML, I18N(국제화), 데이터베이스(SQL), 함수(Function) 라이브러리로 구성됨.
- 자바의 표준 스펙을 주관하는 JCP에서 스펙을 제정하며 현재 1.2스펙까지 나와 있다.

JSTL 라이브러리별 URI 및 prefix

| 라이브러리 | URI | prefix |
|-------------|---|--------|
| Core(핵심) | http://java.sun.com/jsp/jstl/core | c |
| I18N(데이터포맷) | http://java.sun.com/jsp/jstl/fmt | fmt |
| 데이터베이스 | http://java.sun.com/jsp/jstl/sql | sql |
| 함수 | http://java.sun.com/jsp/jstl/functions | fn |
| XML | http://java.sun.com/jsp/jstl/xml | x |



JSTL(JSP 표준태그 라이브러리)

JSTL(Jsp Standart Tag Library) 구성

JSTL이 제공하는 태그의 종류


| 태그 | 기능 |
|---------------|--|
| Core 태그 | <ul style="list-style-type: none">- 변수 선언, 삭제 등 변수와 관련된 작업- if문 for문과 같은 제어기능- URL 처리로 페이지 이동 기능 |
| Formatting 태그 | <ul style="list-style-type: none">- 문자열이나 컬렉션을 처리하는 함수 태그- 숫자, 날짜, 시간 등을 형식화하는 기능- 다국어 지원 기능을 제공 |
| Sql 태그 | <ul style="list-style-type: none">- 데이터베이스와 상호 작용하기 위해 사용하는 태그- 데이터의 삽입, 수정, 삭제, 조회 기능을 제공 |
| Functions 태그 | <ul style="list-style-type: none">- 문자열을 처리하는 함수 제공 |



JSTL(JSP 표준태그 라이브러리)


JSTL(Jsp Standart Tag Library) 설치

- 배포 사이트 : <https://mvnrepository.com/>
- jstl 1.2.jar 다운로드

**JSTL » 1.2**
JSTL

| | |
|--------------|---|
| License | CDDL GPL 2.0 |
| Categories | Java Specifications |
| Date | (Jun 23, 2011) |
| Files | pom (356 bytes) jar (404 KB) View All |
| Repositories | Central Redhat GA WSO2 Dist |
| Used By | 1,739 artifacts |

Jar 파일 선택



▼ JstlExam

- > Deployment Descriptor: JstlExam
- > JAX-WS Web Services
- > Java Resources
- > JavaScript Resources
- > build
- ▼ WebContent
 - > META-INF
 - ▼ WEB-INF
 - lib
 - jstl-1.2.jar



JSTL(JSP 표준태그 라이브러리)

JSTL 태그 사용하기

```
<%@ taglib prefix="태그 식별이름" uri="태그 지원 URL" %>
```



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

Core 태그의 종류

| 태그 | 설명 |
|---------------|----------------------------------|
| <c:set> | 사용할 변수를 설정 |
| <c:remove> | 설정된 변수를 제거 |
| <c:if> | 조건문 처리 |
| <c:forEach> | 반복문 처리 |
| <c:choose> | 다중 조건문 처리 |
| <c:when> | <c:choose>의 서브 태그로 조건문이 참일 때 수행 |
| <c:otherwise> | <c:choose>의 서브 태그로 조건문이 거짓일 때 수행 |
| <c:out> | 출력에 사용 |



JSTL(JSP 표준태그 라이브러리)

core 태그 - <c:if> 태그 실습

<c:if test="\${조건식}">출력문</c:if>

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>core 예제</title>
</head>
<body>
    <c:set var="num" value="11" /> <!-- 변수 선언 -->
    <c:if test="${num%2==0}">
        ${num}은 짝수 입니다.
    </c:if>
    <c:if test="${num%2==1}">
        ${num}은 홀수 입니다.
    </c:if>
</body>
```

core01.jsp



JSTL(JSP 표준태그 라이브러리)

core 태그 - <c:if> 태그 실습

- 홀수/짝수 판별하는 프로그램 : 입력 폼 만들기

홀수/짝수 판별 프로그램

숫자 :

core02.jsp

```
<h2>홀수/짝수 판별 프로그램</h2>
<form action="core02_process.jsp" method="get">
  <p>숫자 : <input type="text" name="number">
    <input type="submit" value="전송"></p>
</form>
```



JSTL(JSP 표준태그 라이브러리)

core 태그 - <c:if> 태그 실습

```
<title>core 예제</title>
</head>
<body>
    <!-- <%
        String number = request.getParameter("number");
    %>
    <c:set var="num" value="<%=number %>" /> --%>

    <c:set var="num" value="${param.number}" />
    <c:if test="${num%2==0}">
        ${num}은 짝수 입니다.
    </c:if>
    <c:if test="${num%2==1}">
        ${num}은 홀수 입니다.
    </c:if>
</body>
```



JSTL(JSP 표준태그 라이브러리)

<c:choose> 태그

```
<c:choose>  
  <c:when> test="조건식1">본문 내용1</c:when>  
  <c:when> test="조건식1">본문 내용1</c:when>  
  ....  
  <c:otherwise> 본문내용 <c:/otherwise>  
</c:choose>
```



JSTL(JSP 표준태그 라이브러리)

<c:choose> 태그

← → ↻ ⓘ localhost:8181/Chapter14/scoreForm.jsp

점수를 입력해 주세요

점수 :

시험 점수 : 82

B학점입니다.

```
<h2>점수를 입력해 주세요</h2>
<form action="scoreTest.jsp" method="get">
  <p>점수 : <input type="text" name="score">
    <input type="submit" value="학점출력"></p>
</form>
```

JSTL(JSP 표준태그 라이브러리)

<c:choose> 태그

```
<c:set var="score" value="${param.score}" />
<h2>시험 점수 : ${score}</h2>
<c:choose>
  <c:when test="${score}>=90 && score<=100}">
    <h3>A학점입니다.</h3>
  </c:when>
  <c:when test="${score}>=80 && score<90}">
    <h3>B학점입니다.</h3>
  </c:when>
  <c:when test="${score}>=70 && score<80}">
    <h3>C학점입니다.</h3>
  </c:when>
  <c:when test="${score}>=60 && score<70}">
    <h3>D학점입니다.</h3>
  </c:when>
  <c:otherwise>
    <h3>F학점입니다.</h3>
  </c:otherwise>
</c:choose>
```



JSTL(JSP 표준태그 라이브러리)

<c:forEach> 태그

<c:forEach **var**=변수이름" **items**=“반복할 객체이름” **begin**=“시작값”
end= “ 마지막값” **step**=“증가값”>

구구단 출력

| | | | | | | | | |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1x2=2 | 2x2=4 | 3x2=6 | 4x2=8 | 5x2=10 | 6x2=12 | 7x2=14 | 8x2=16 | 9x2=18 |
| 1x3=3 | 2x3=6 | 3x3=9 | 4x3=12 | 5x3=15 | 6x3=18 | 7x3=21 | 8x3=24 | 9x3=27 |
| 1x4=4 | 2x4=8 | 3x4=12 | 4x4=16 | 5x4=20 | 6x4=24 | 7x4=28 | 8x4=32 | 9x4=36 |
| 1x5=5 | 2x5=10 | 3x5=15 | 4x5=20 | 5x5=25 | 6x5=30 | 7x5=35 | 8x5=40 | 9x5=45 |
| 1x6=6 | 2x6=12 | 3x6=18 | 4x6=24 | 5x6=30 | 6x6=36 | 7x6=42 | 8x6=48 | 9x6=54 |
| 1x7=7 | 2x7=14 | 3x7=21 | 4x7=28 | 5x7=35 | 6x7=42 | 7x7=49 | 8x7=56 | 9x7=63 |
| 1x8=8 | 2x8=16 | 3x8=24 | 4x8=32 | 5x8=40 | 6x8=48 | 7x8=56 | 8x8=64 | 9x8=72 |
| 1x9=9 | 2x9=18 | 3x9=27 | 4x9=36 | 5x9=45 | 6x9=54 | 7x9=63 | 8x9=72 | 9x9=81 |

JSTL(JSP 표준태그 라이브러리)

- 구구단 출력하기 : 스크립트 vs JSTL

```
<table>
  <% for(int i=2; i<10; i++){ %>
  <tr>
    <% for(int j=1; j<10; j++){ %>
      <td width="80"><%=j + "x" + i + "=" + (i*j)%></td>
    <% } %>
  <tr>
  <%} %>
</table>
```

Core03.jsp

```
<table>
  <c:forEach var="i" begin="1" end="9">
  <tr>
    <c:forEach var="j" begin="2" end="9">
      <td width="100">${j} x ${i} = ${i*j}</td>
    </c:forEach>
  </tr>
  </c:forEach>
</table>
```

Core03_2.jsp



JSTL(JSP 표준태그 라이브러리)

JSTL 실습

- 예제는 Product 클래스를 만들어 JSP에서 <jsp:useBean> 액션과 표현 언어를 사용하는 구조로 구성되어 있다.

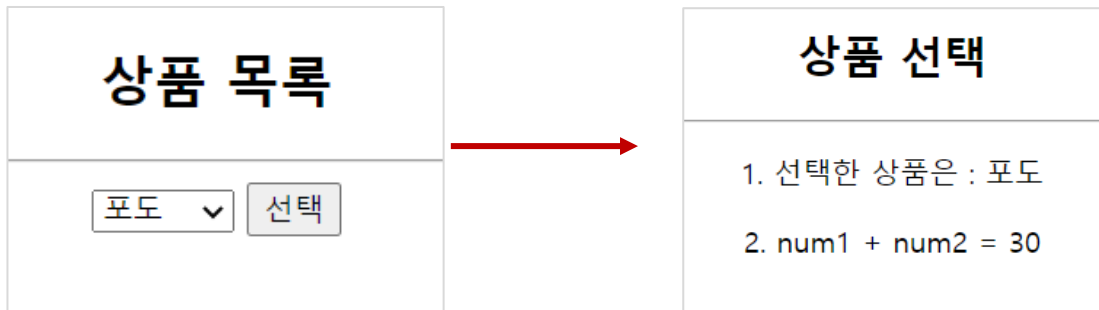
프로그램 소스 목록

| 파일 이름 | 역 할 |
|-----------------|--|
| Product.java | 상품 정보를 제공하는 빈즈클래스로 jsp에 데이터를 공급 |
| productList.jsp | 상품 목록을 출력하기 위한 jsp 파일 |
| productSel.jsp | productList에서 item을 선택하고 <확인> 버튼을 누르면 호출되는 jsp로 표현언어를 이용해 데이터 출력 |



JSTL(JSP 표준태그 라이브러리)

JSTL 실습



```
package org.bean;

public class Product {
    private String[] productList = {"귤", "사과", "포도", "토마토", "감"};
    private int num1 = 10;
    private int num2 = 20;

    public int getNum1() {
        return num1;
    }

    public int getNum2() {
        return num2;
    }

    public String[] getProductList() {
        return productList;
    }
}
```

Product.java



JSTL(JSP 표준태그 라이브러리)

JSTL 실습

```
<body>
<jsp:useBean id="product" class="com.beans.Product" />
  <div id="container">
    <h2>상품 목록</h2>
    <hr>
    <form action="productSel.jsp" method="post">
      <select name="select">
        <%-- <%
          for(String item : product.getProductList()){
            out.println("<option>" + item + "</option>");
          }
        %> --%>
        <c:forEach var="item" items="${product.productList}" >
          <option>${item}</option>
        </c:forEach>
      </select>
      <input type="submit" value="선택">
    </form>
  </div>
</body>
```

productList.jsp



EL 언어

JSTL 실습

productSel.jsp

```
<title>EL 예제</title>
<style type="text/css">
    #container{width: 100%; margin: 0 auto; text-align: center;}
</style>
</head>
<body>
    <div id="container">
        <h2>상품 선택</h2>
        <hr>
        <p>1. 선택한 상품은 : ${param.select}
        <p>2. num1 + num2 = ${product.num1 + product.num2}
    </div>
</body>
```



JSTL(JSP 표준태그 라이브러리)

core 태그 실습 예제

- 이미지 리스트 출력하기

| 이미지 | 이미지 이름 | 선택하기 |
|--|--------------|--------------------------|
|  | 이미지 이름 : 커피1 | <input type="checkbox"/> |
|  | 이미지 이름 : 커피2 | <input type="checkbox"/> |
|  | 이미지 이름 : 커피3 | <input type="checkbox"/> |

JSTL(JSP 표준태그 라이브러리)

```
<ul class="list">
  <li>
    <span style="margin-left:50px">이미지</span>
    <span>이미지 이름</span>
    <span>선택하기</span>
  </li>
  <c:forEach var="i" begin="1" end="9" step="1">
    <li>
      <a href="#" style="margin-left:50px">
        </a>
        <a href="#"><strong>이미지 이름 : 커피${i}</strong></a>
        <a href="#"><input name="chk${i}" type="checkbox"></a>
      </li>
    </c:forEach>
  </ul>
  <style type="text/css">
    .list{width:80%; padding:0 10px 10px; margin:0 auto}
    .list li{margin: 10px 0; line-height: 100px;
      list-style:none ; border-bottom: 2px solid #ccc; position:releative;}
    .list li img{float:left; position:absolute; }
    .list li a{text-decoration:none; margin-left:340px}
    .list li a:hover{text-decoration:underline}
    .list li span{color:blue; margin-left:320px; font-size:14px;}
  </style>
```



JSTL(JSP 표준태그 라이브러리)

- MVC 구현시 <c:if> 사용

```
<body>
  <c:if test="${msg eq 'update' }">
    <script type="text/javascript">
      alert('게시글이 수정되었습니다. ');
      history.go(-1);
    </script>
  </c:if>
  <!-- <c:if test="${msg eq 'delete' }">
    <script type="text/javascript">
      alert('게시글이 삭제되었습니다. ');
      history.go(-1);
    </script>
  </c:if> --%>
  <h2>게시판 목록</h2>
</body>
```



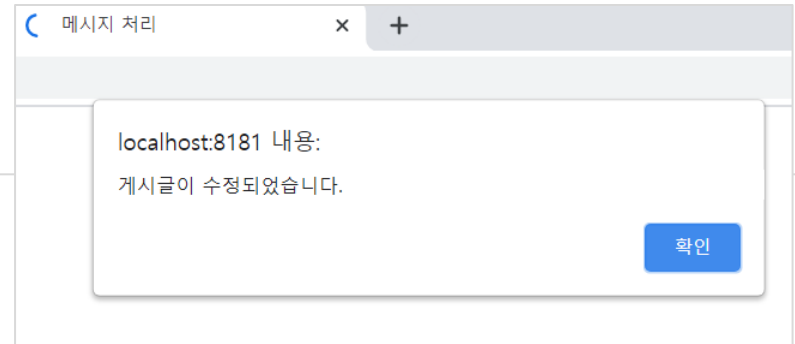
JSTL(JSP 표준태그 라이브러리)

- MVC 구현시 <c:if> 사용

```
@WebServlet("/message")
public class MsgServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        //model 저장
        request.setAttribute("msg", "update");
        request.setAttribute("msg", "delete");

        //view
        RequestDispatcher dispatcher = request.getRequestDispatcher("message.jsp");
        dispatcher.forward(request, response);
    }
}
```



JSTL fmt 태그

core 태그

- <c:url> 태그

URL에 자동으로 **Context Path**(웹 프로젝트 이름)를 붙여주는 일을 한다.

Context 이름이 WebMarket라면 ->

WebMarket/member/loginMember.jsp 또는 WebMarket/products.jsp

```
<c:choose>
  <c:when test="${empty sessionId}">
    <li class="nav-item"><a href="<c:url value="/member/loginMember.jsp"/>" class="nav-link">로그인</a></li>
    <li class="nav-item"><a href="<c:url value="/member/addMember.jsp"/>" class="nav-link">회원 가입</a></li>
  </c:when>
  <c:otherwise>
    <li style="padding-top: 7px; color: white">[<%=sessionId %>]님</li>
    <li class="nav-item"><a href="<c:url value="/member/logoutMember.jsp"/>" class="nav-link">로그 아웃</a></li>
    <li class="nav-item"><a href="<c:url value="/member/updateMember.jsp"/>" class="nav-link">회원 수정</a></li>
  </c:otherwise>
</c:choose>
<li class="nav-item"><a href="<c:url value="/products.jsp" />" class="nav-link">상품 목록</a></li>
<li class="nav-item"><a href="<c:url value="/addProduct.jsp" />" class="nav-link">상품 등록</a></li>
```



JSTL fmt 태그

formatting 태그

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

사용자의 로케일(Locale)에 따라 숫자를 다양한 형식으로 출력하기

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

```
<p> 숫자 : <fmt:formatNumber value="3200100" />  
<p><fmt:formatNumber value="3200100" type="number" />  
<p><fmt:formatNumber value="3200100" groupingUsed="false" />  
<p><fmt:formatNumber value="3200100" groupingUsed="true" />  
<p><fmt:formatNumber value="3200100" type="currency" currencySymbol="\$"/>  
<p><fmt:formatNumber value="0.25" type="percent"/>  
<p><fmt:formatNumber value="3200100.25" pattern="#,##0.0"/>
```

숫자 : 3,200,100

3,200,100

3200100

3,200,100

\$3,200,100.00

25%

3,200,100.2



fmt 태그를 이용한 다국어 처리

사용자의 로케일에 따라 날짜/타임존 출력

```
<jsp:useBean id="now" class="java.util.Date" />
<body>
  <h2>날짜를 다양한 형식으로 표기</h2>
  <p>현재 날짜 및 시간 : ${now}</p>
  <p><fmt:formatDate value="${now}" type="date" /></p>
  <p><fmt:formatDate value="${now}" type="time" /></p>
  <p><fmt:formatDate value="${now}" type="both" /></p>
  <p><fmt:formatDate value="${now}" pattern="yyyy:MM:dd"/></p>
  <p><fmt:formatDate value="${now}" pattern="yyyy:MM:dd hh:mm:ss"/></p>

  <p>== 타임존 출력 ==</p>
  <p><fmt:timeZone value="Europe/London">
    런던 : <fmt:formatDate value="${now}" type="both" dateStyle="full" timeStyle="full"/>
  </fmt:timeZone>
  <p><fmt:timeZone value="America/New_York">
    뉴욕 : <fmt:formatDate value="${now}" type="both" dateStyle="full" timeStyle="full"/>
  </fmt:timeZone>
</body>
```

날짜를 다양한 형식으로 표기

현재 날짜 및 시간 : Mon Mar 22 05:47:35 KST 2021

2021. 3. 22.

오전 5:47:35

2021. 3. 22. 오전 5:47:35

2021:03:22

2021:03:22 05:47:35

== 타임존 출력 ==

런던 : 2021년 3월 21일 일요일 오후 8시 47분 35초 그리니치 표준시

뉴욕 : 2021년 3월 21일 일요일 오후 4시 47분 35초 미 동부 하계 표준시



다국어 처리

다국어 처리

- JSP 페이지에 JSTL의 `fmt` 태그를 이용하면 언어별로 페이지를 따로 만들 필요 없이 아주 간단하게 다국어를 지원할 수 있다.
- 지역화는 사용 국가별 환경에서 특정 언어와 지역에 맞게 적합화 하는 것으로 `L10n`으로 표기
 - 숫자, 날짜, 시간의 형식
 - 화폐의 표시
 - 키보드의 지원 등
- 국제화는 여러 국가에서 사용할 수 있도록 다국어를 지원하는 것으로 `i18n`으로 표기
 - 유니코드의 사용이나 기존의 인코딩을 적절히 처리해야 한다.
 - 날짜와 시간 표시, 지역의 달력, 숫자 표시등 사용자 설정을 지원해야 한다.



다국어 처리

Locale 클래스를 이용한 다국어 처리

- 웹 페이지에 보이는 메시지가 여러 가지 언어로 주어졌을 때 사용자가 어떤 언어로 출력할 것인지 결정 할 수 있게 하는 클래스이다.
- Locale 클래스를 사용하려면 JSP 페이지에 page 디렉티브
- 태그의 import 속성으로 java.util.Locale을 설정해야 한다.

Locale locale = request.getLocale();

로케일 감지 메소드의 종류

| 메소드 | 반환 유형 | 설명 |
|---------------------|--------|-------------------------|
| getCountry() | String | 현재 로케일의 국가/지역 코드를 가져온다. |
| getLanguage() | String | 현재 로케일의 언어 코드를 가져온다. |
| getDisplayCountry() | String | 현재 로케일의 국가 이름을 가져온다. |

Module java.base

Package java.util

Class Locale

java.lang.Object
java.util.Locale

All Implemented Interfaces:

Serializable, Cloneable

```
public final class Locale
extends Object
implements Cloneable, Serializable
```



다국어 처리

사용자의 로케일에 따라 국가, 날짜, 통화 형식 출력하기

```
<%@ page import="java.util.Locale, java.util.Date" %>
<%@ page import="java.text.DateFormat, java.text.NumberFormat" %>
```

```
<%
    Locale locale = request.getLocale(); //지역
    Date currentDate = new Date(); //날짜

    //날짜 형식
    DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.FULL, locale);
    //숫자 형식
    NumberFormat numberFormat = NumberFormat.getNumberInstance(locale);
%>
<body>
    <h2>현재 로케일의 국가, 날짜, 숫자</h2>
    <p>국가 : <%=locale.getDisplayCountry() %>
    <p>언어 : <%=locale.getDisplayLanguage() %>
    <p>날짜 : <%=currentDate %>
    <p>날짜2 : <%=dateFormat.format(currentDate) %>
    <p>숫자(12345.67) : <%=numberFormat.format(12345.67) %>
</body>
```

현재 로케일의 국가, 날짜, 통화

국가 :

언어 : 한국어

날짜 : 2020년 11월 6일 금요일

숫자(12345.67): 12,345.67

fmt 태그를 이용한 다국어 처리

JSTL fmt 태그를 이용한 다국어 처리

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

주요 태그의 기능과 사용법

리소스 번들

리소스 번들은 메시지 처리 태그에서 사용하는 파일로 java.util.Properties 클래스에서 정의된 방법으로 확장자가 properties인 파일이 반드시 있어야 함
알파벳이나 숫자, 라틴 문자 외의 언어를 유니코드 값으로 표현한다.

| *.properties 파일 | 설명 |
|--------------------|-------------|
| 파일이름.properties | 기본 메시지일때 사용 |
| 파일이름_ko.properties | 한글 메시지일때 사용 |
| 파일이름_en.properties | 영어 메시지일때 사용 |

Module java.base

Package java.util

Class Properties

java.lang.Object

java.util.Dictionary<K,V>

java.util.Hashtable<Object,Object>

java.util.Properties

All Implemented Interfaces:

Serializable, Cloneable, Map<Object,Object>



fmt 태그를 이용한 다국어 처리

setLocale 태그

setLocale 태그는 국제화 태그가 사용할 로케일을 설정하는 태그이다.

다국어를 지원하는 웹 페이지를 만들 때 리소스번들인 *.properties 파일과 연계 사용함

```
<fmt:setLocale value= "언어 코드"/>
```

bundle 태그

Bundle 태그는 사용할 리소스 번들을 설정하는 태그이다.

```
<fmt:setBundle basename= "리소스번들"/>
```

message 태그

message 태그는 bundle 태그에 설정한 리소스 번들에서 메시지를 읽어와 출력하는 태그이다.

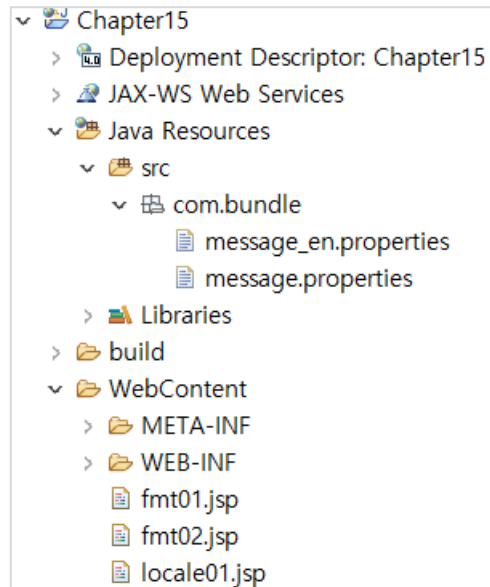
```
<fmt:message key= "key 이름"/>
```



fmt 태그를 이용한 다국어 처리

사용자의 로케일에 따라 리소스 번들의 메시지 출력하기

- ▶ src 하위에 com.bundle 패키지 생성 > properties 파일 만들기



message.properties

```
title=\uC790\uB9AC \uC11C\uB844 \uD398\uC774\uC9C0
username = \uAD00\uB9AC\uC790
password = 1234
```

Message_en.properties

```
title=Java Server Page
username=admin
password=1234
```

fmt 태그를 이용한 다국어 처리

사용자의 로케일에 따라 리소스 번들의 메시지 출력하기

← → ↻ ⓘ localhost:8181/Chapter15/fmt01.jsp

--- 기본 로케일 ----
제 목 : 자바 서버 페이지
이 름 : 관리자

--- 영문 로케일 ----
제 목 : Java Server Page
이 름 : admin

```
<p>--- 기본 로케일 ----  
    <fmt:setLocale value="ko"/>  
    <fmt:setBundle basename="com.bundle.message"/>  
<p>제 목 : <fmt:message key="title"/>  
<p>이 름 : <fmt:message key="username" />  
  
<p>--- 영문 로케일 ----  
    <fmt:setLocale value="en"/>  
    <fmt:setBundle basename="com.bundle.message_en"/>  
<p>제 목 : <fmt:message key="title" />  
<p>이 름 : <fmt:message key="username" />
```



fmt 태그를 이용한 다국어 처리

실습 예제 : 링크를 클릭하여 한국어, 영어로 출력하기



fmt 태그를 이용한 다국어 처리

사용자의 로케일에 따라 리소스 번들의 메시지 출력하기

```
<title>다국어 처리</title>
<style type="text/css">
a{text-decoration:none; color:#000;}
a:hover{text-decoration: underline; color:#333;}
</style>
</head>
<body>
    <h2>Java Server Page</h2>

    <fmt:setLocale value="${param.language}"/>
    <fmt:setBundle basename="com.bundle.message"/>

    <a href="?language=ko">한국어</a> | <a href="?language=en">영어</a>

    <p>제 목 : <fmt:message key="title" />
    <p>이 름 : <fmt:message key="username"/>
</body>
```



JSTL Functions 태그

functions 태그

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

- <fn:contains>와 <fn:containsIgnoreCase> 태그로 문자열 검색하기

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

```
<body>
  <p>java 문자열 검색
  <p>Hello, Java Server Pages! =>
    ${fn:contains("Hello, Java Server Pages!", "java")}
  <p>Hello, Java Server Pages! =>
    ${fn:containsIgnoreCase("Hello, Java Server Pages!", "java")}

  <!-- contains() -> 대, 소문자 구분하여 포함되었는지 확인
       containsIgnoreCase() -> 대, 소문자 구분하지 않음
</body>
```

java 문자열 검색

Hello, Java Server Pages! => false

Hello, Java Server Pages! => true



JSTL Functions 태그

functions 태그 실습 예제

- <fn:split>와 <fn:join> 태그로 문자열 분리하고 연결하기

```
<h2>Java 문자열 분리, 연결</h2>
<p>Hello, Java Server Page! =>
<p><c:set var="text" value="${fn:split('Hello, Java Server Page!', ' ')}" />
<p>text[0] = ${text[0]}
<p>text[1] = ${text[1]}
<p>문자열의 개수 : ${fn:length(text)}

<p><c:forEach var="i" begin="0" end="${fn:length(text)-1}"
    text[${i}] = ${text[i]}<br>
</c:forEach>

<p>문자열 연결 : 구분 기호(문자)로 연결하기
<p>${fn:join(text, '-')} }
```

Java 문자열 분리, 연결

Hello, Java Server Page! =>

text[0] = Hello,

text[1] = Java

문자열의 개수 : 4

text[0] = Hello,

text[1] = Java

text[2] = Server

text[3] = Page!

문자열 연결 : 구분 기호(문자)로 연결하기

Hello,-Java-Server-Page!



JSTL sql 태그

sql 태그

Jsp에서 데이터베이스 를 사용할수 있도록 기능을 만들어 둔 커스텀 태그이다.

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

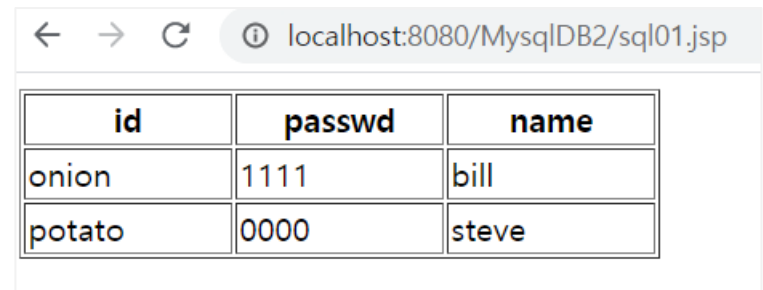
| 태그 | 기능 |
|---------------------|--------------------------|
| <sql:setDataSource> | DataSource를 설정하는데 사용 |
| <sql:query> | 조회 쿼리문을 실행하는데 사용 |
| <sql:update> | 삽입, 수정, 삭제 쿼리문을 실행하는데 사용 |
| <sql:param> | 폼페이지에서 전송된 name 값 설정 |



JSTL sql 태그

<sql:query> 태그로 SELECT 쿼리문 실행하기.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSTL</title>
</head>
<body>
    <!-- DataSource - DB연결 -->
    <sql:setDataSource var="dataSource"
        driver = "com.mysql.jdbc.Driver"
        url = "jdbc:mysql://localhost:3306/jspdb?useUnicode=true&characterEncoding=utf8"
        user = "root" password="12345"
    />
```



| id | passwd | name |
|--------|--------|-------|
| onion | 1111 | bill |
| potato | 0000 | steve |



JSTL sql 태그

<sql:query>태그로 SELECT 쿼리문 실행하기.

```
<!-- member 테이블 조회 -->
<sql:query var="resultSet" dataSource="${dataSource}">
    SELECT * FROM member;
</sql:query>

<table border="1">
    <tr>
        <c:forEach var="columnName" items="${resultSet.columnNames}">
            <th width="100"><c:out value="${columnName }" /></th>
        </c:forEach>
    </tr>
    <c:forEach var="row" items="${resultSet.rowsByIndex}">
        <tr>
            <c:forEach var="column" items="${row }" varStatus="i">
                <td>
                    <c:if test="${column != null }">
                        <c:out value="${column }" />
                    </c:if>
                    <c:if test="${column == null }">
                        &nbsp;
                    </c:if>
                </td>
            </c:forEach>
        </tr>
    </c:forEach>
</tr>
```



JSTL sql 태그

<sql:update> 태그로 INSERT 쿼리문 실행하기.

회원 가입

아 이 디:

패스워드:

이 름:

→

| id | passwd | name |
|--------|--------|-------|
| apple | 12345 | 전사과 |
| onion | 1111 | bill |
| potato | 0000 | steve |

```
<h2>회원 가입</h2>
<hr>
<form action="insert02_process.jsp" method="post">
  <p>아 이 디: <input type="text" name="id">
  <p>패스워드: <input type="password" name="passwd">
  <p>이 름: <input type="text" name="name">
  <p><input type="submit" value="전송">
</form>
```



JSTL sql 태그

<sql:update> 태그로 INSERT 쿼리문 실행하기.

```
<%  
    String id = request.getParameter("id");  
    String passwd = request.getParameter("passwd");  
    String name = request.getParameter("name");  
%>  
<sql:setDataSource var="dataSource"  
    driver = "com.mysql.jdbc.Driver"  
    url = "jdbc:mysql://localhost:3306/jspdb?useUnicode=true&characterEncoding=utf8"  
    user = "root" password="12345"  
/>  
  
<!-- member 테이블에 데이터 삽입 -->  
<sql:update dataSource="${dataSource}" var="resultSet">  
    INSERT INTO member(id, passwd, name) VALUES (?, ?, ?)  
    <sql:param value="<%=id %>" />  
    <sql:param value="<%=passwd %>" />  
    <sql:param value="<%=name %>" />  
</sql:update>  
  
<!-- member 테이블 조회 -->  
<c:import var="url" url="sql01.jsp" />  
${url}
```


JSTL sql 태그

<sql:update> 태그로 UPDATE 쿼리문 실행하기.

회원 정보 수정

아 이 디:

패스워드:

이 름:

회원 정보 수정

| id | passwd | name |
|--------|--------|-------|
| apple | 12345 | 전사과 |
| onion | 1111 | 이양파 |
| potato | 0000 | steve |

```
<h2>회원 정보 수정</h2>
<hr>
<form action="sql03_process.jsp" method="post">
  <p>아 이 디: <input type="text" name="id">
  <p>패스워드: <input type="password" name="passwd">
  <p>이 름: <input type="text" name="name">
  <p><input type="submit" value="전송">
</form>
```



JSTL sql 태그

<sql:update> 태그로 UPDATE 쿼리문 실행하기.

```
<%
    String id = request.getParameter("id");
    String passwd = request.getParameter("passwd");
    String name = request.getParameter("name");
%>
<sql:setDataSource var="dataSource"
    driver = "com.mysql.jdbc.Driver"
    url = "jdbc:mysql://localhost:3306/jspdb?useUnicode=true&characterEncoding=utf8"
    user = "root" password="12345"
/>

<!-- member 테이블 데이터 수정 -->
<sql:update dataSource="${dataSource}" var="resultSet">
    UPDATE member SET name = ? WHERE id = ? and passwd = ?
    <sql:param value="<%=name %>" />
    <sql:param value="<%=id %>" />
    <sql:param value="<%=passwd %>" />
</sql:update>

<!-- member 테이블 조회 -->
<c:import var="url" url="sql01.jsp" />
${url}
```



JSTL sql 태그

`<sql:update>` 태그로 DELETE 쿼리문 실행하기.

회원 정보 삭제

아 이 디:

패스워드:

회원 정보 삭제

| id | passwd | name |
|-------|--------|------|
| apple | 12345 | 전사과 |
| onion | 1111 | 이양파 |

```
<h2>회원 정보 삭제</h2>
<hr>
<form action="sql04_process.jsp" method="post">
  <p>아 이 디: <input type="text" name="id">
  <p>패스워드: <input type="password" name="passwd">
  <p><input type="submit" value="전송">
</form>
```



JSTL sql 태그

<sql:update> 태그로 DELETE 쿼리문 실행하기.

```
<%
    String id = request.getParameter("id");
    String passwd = request.getParameter("passwd");
    String name = request.getParameter("name");
%>
<sql:setDataSource var="dataSource"
    driver = "com.mysql.jdbc.Driver"
    url = "jdbc:mysql://localhost:3306/jspdb?useUnicode=true&characterEncoding=utf8"
    user = "root" password="12345"
/>

<!-- member 테이블 데이터 삭제 -->
<sql:update dataSource="${dataSource}" var="resultSet">
    DELETE FROM member WHERE id = ? and passwd = ?
    <sql:param value="<%=id %>" />
    <sql:param value="<%=passwd %>" />
</sql:update>

<!-- member 테이블 조회 -->
<c:import var="url" url="sql01.jsp" />
${url}
```

