

9 다국어 처리

1

다국어 처리의 개요

2

Locale 클래스를 이용한 다국어 처리

3

JSTL fmt 태그를 이용한 다국어 처리

4

[웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

학습목표

- 다국어 처리의 개념을 이해합니다.
- Locale 클래스를 이용한 다국어 처리 방법을 익힙니다.
- JSTL fmt 태그를 이용한 다국어 처리 방법을 익힙니다.
- 다국어 처리를 이용하여 한글/영문 상품 등록 페이지를 만듭니다.

1. 다국어 처리의 개요

❖ 다국어 처리

- 웹 브라우저를 사용하는 국가에 따라 다양한 언어 및 지역을 지원하는 서비스
- 다른 언어와 지역적 차이를 기술 변경 없이 소프트웨어에 바로 적용하는 것
- JSP 페이지에 JSTL의 fmt 태그를 이용하면 언어별로 페이지를 따로 만들 필요 없이 아주 간단하게 다국어를 지원할 수 있음
- 다국어는 다양한 언어와 지역에 적용될 수 있도록 하는 국제화(internationalization, i18n)와 언어별 구성 요소를 추가하여 특정 지역의 언어나 문화에 맞추는 지역화(localization, L10n)를 포함

1. 다국어 처리의 개요

❖ 지역화

- 사용 국가별 환경에서 특정 언어와 지역에 맞게 적합화하는 것
- L10n으로 표기

❖ 지역화에 주로 고려되는 사항

- 숫자, 날짜, 시간의 형식
- 화폐의 표시
- 키보드의 지원
- 문자열의 순서와 정렬
- 심벌, 아이콘, 색상
- 문화에 따라 오해의 소지가 있거나 의미가 없는 문자, 그림
- 지역별 법률의 차이

1. 다국어 처리의 개요

❖ 1.2 국제화

- 여러 국가에서 사용할 수 있도록 다국어를 지원하는 것
- i18n으로 표기
- 국제화는 어느 국가에서나 사용할 수 있게 하는 지역화 기능을 포함
- 국제화는 주로 다음과 같은 처리를 포함하여 지원해야 함
 - 국제화 지원은 유니코드의 사용이나 기존의 인코딩을 적절히 처리하고 사용자 인터페이스에 표시할 문자열에는 문자 코드가 포함되지 않도록 설계 및 개발해야 함
 - 국제화를 처리하는 정보에 언어 정보를 포함하거나, 세로쓰기/가로쓰기/우측에서 좌측으로의 가로쓰기 등 언어의 특성을 반영하는 처리 등을 지원해야 함
 - 날짜와 시간 표시, 지역의 달력, 숫자 표시, 리스트의 정렬과 표시, 인명이나 주소의 처리 등 언어의 특성(언어적·지역적·문화적 특성 등)에 대한 사용자 설정 지원해야 함
 - 국제화는 사용자의 요청이나 설정에 따라 필요시 사용되도록 지역화 정보를 코드와 분리해야 함

2. Locale 클래스를 이용한 다국어 처리

❖ Locale 클래스

- 특정 지리적·정치적·문화적 지역을 나타내는 클래스
- 사용자의 지역 환경에 따라 결정되는 지역적 문화(언어, 날짜, 시간 등)의 정보를 담고 있음
 - 예를 들어 웹 페이지에 보이는 메시지가 여러 가지 언어로 주어졌을 때 사용자가 어떤 언어로 출력할 것인지 결정할 수 있게 하는 수단이 바로 Locale 클래스
- Locale 클래스는 단순한 메시지뿐 아니라 숫자, 날짜, 시간 등을 표현하는 데 사용
- Locale 객체의 생성은 request 내장 객체를 이용하여 현재 웹 브라우저에 미리 정의된 언어나 국가 정보를 가져오는 방법

```
java.util.Locale request.getLocale();
```

- Locale 클래스를 사용하려면 JSP 페이지에 page 디렉티브 태그의 import 속성으로 패키지 java.util.Locale을 설정

2. Locale 클래스를 이용한 다국어 처리

❖ 로케일 감지하기

- 웹 브라우저에 설정된 국가와 언어 이름을 알아내는 것을 로케일 감지

❖ 로케일 감지 메소드의 종류

메소드	반환 유형	설명
getDefault()	static Locale	디폴트 로케일의 현재 값을 가져옵니다.
getCountry()	String	현재 로케일의 국가/지역 코드(대문자)를 가져옵니다.
getDisplayCountry()	String	현재 로케일의 국가 이름을 가져옵니다.
getLanguage()	String	현재 로케일의 언어 코드(소문자)를 가져옵니다.
getDisplayLanguage()	String	현재 로케일의 언어 이름을 얻어옵니다.

2. Locale 클래스를 이용한 다국어 처리

[로케일 감지 메소드 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Locale"%>
<%
    Locale locale = request.getLocale();
    String displayLanguage = locale.getDisplayLanguage();
    String language = locale.getLanguage();
    String displayCountry = locale.getDisplayCountry();
    String country = locale.getCountry();
%>
<html>
<head>
<title>Localization</title>
</head>
<body>
    <%
        out.println("로케일 언어 : " + displayLanguage + "<br />");
        out.println("로케일 언어 코드 : " + language + "<br />");
        out.println("로케일 국가 : " + displayCountry + "<br />");
        out.println("로케일 국가 코드 : " + country + "<br />");
    %>
</body>
</html>
```

로케일 언어 : 한국어
로케일 언어 코드 : ko
로케일 국가 : 대한민국
로케일 국가 코드 : KR

2. Locale 클래스를 이용한 다국어 처리

❖ 로케일 표현하기

■ 언어 설정

- 다양한 국가별 언어를 제대로 표현하기 위해 response 내장 객체의 setHeader() 메소드 사용

[로케일의 언어 설정 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Locale"%>
<%
    response.setContentType("text/html");
    response.setHeader("Content-Language", "es");
    String title = "Español";
%>
<html>
<head>
<title>
    <% out.print(title); %>
</title>
</head>
<body>
    <p>Idioma : Español</p>
    <p>¡Hola Mundo!</p>
</body>
</html>
```

Idioma : Español

¡Hola Mundo!

2. Locale 클래스를 이용한 다국어 처리

❖ 2.2 로케일 표현하기

■ 날짜와 시간 설정

- 사용자의 로케일에 따라 특정 날짜와 시간 형식을 표현하기 위해 DateFormat 클래스의 getDateTimelInstance() 메소드를 사용

[로케일의 날짜와 시간 설정 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Locale,java.util.Date"%>
<%@ page import="java.text.DateFormat"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <%
        Locale locale = request.getLocale();
        String date = DateFormat.getDateTimelInstance(DateFormat.FULL,
            DateFormat.SHORT, locale).format(new Date());
    %>
    <p>로케일의 날짜 형식
    <p> <% out.print(date); %>
</body>
</html>
```

로케일의 날짜 형식

2018년 9월 5일 수요일 오후 4:52

2. Locale 클래스를 이용한 다국어 처리

❖ 2.2 로케일 표현하기

■ 통화와 숫자 설정

- 사용자의 로케일에 따라 특정 통화와 숫자를 표현하기 위해 NumberFormat 클래스의 `getCurrencyInstance()` 메소드를 사용

[로케일의 통화와 숫자 설정 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Locale,java.util.Date"%>
<%@ page import="java.text.NumberFormat"%>
<html>
<head>

<title>Internationalization</title>
</head>
<body>
    <%
        Locale locale = request.getLocale();
        NumberFormat curency = NumberFormat.getCurrencyInstance(locale);
        NumberFormat percentage = NumberFormat.getPercentInstance(locale);
        String fcurency = curency.format(1234567);
        String fpercentage = percentage.format(0.25);
    %>
    <p> 로케일의 통화 형식 : <% out.print(fcurency); %>
    <p> 로케일의 비율 형식 : <% out.print(fpercentage); %></p>
</body>
</html>
```

로케일의 통화 형식 : ₩1,234,567

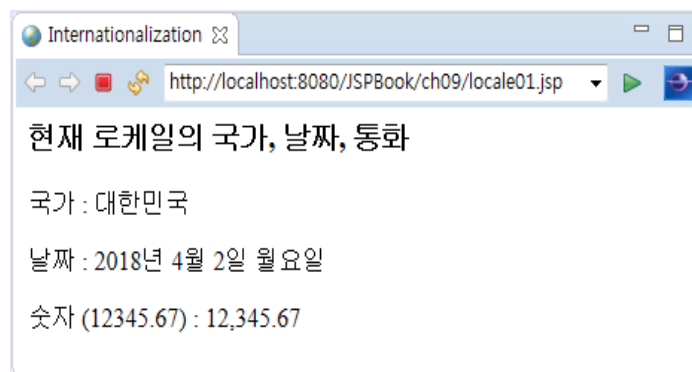
로케일의 비율 형식 : 25%

2. Locale 클래스를 이용한 다국어 처리

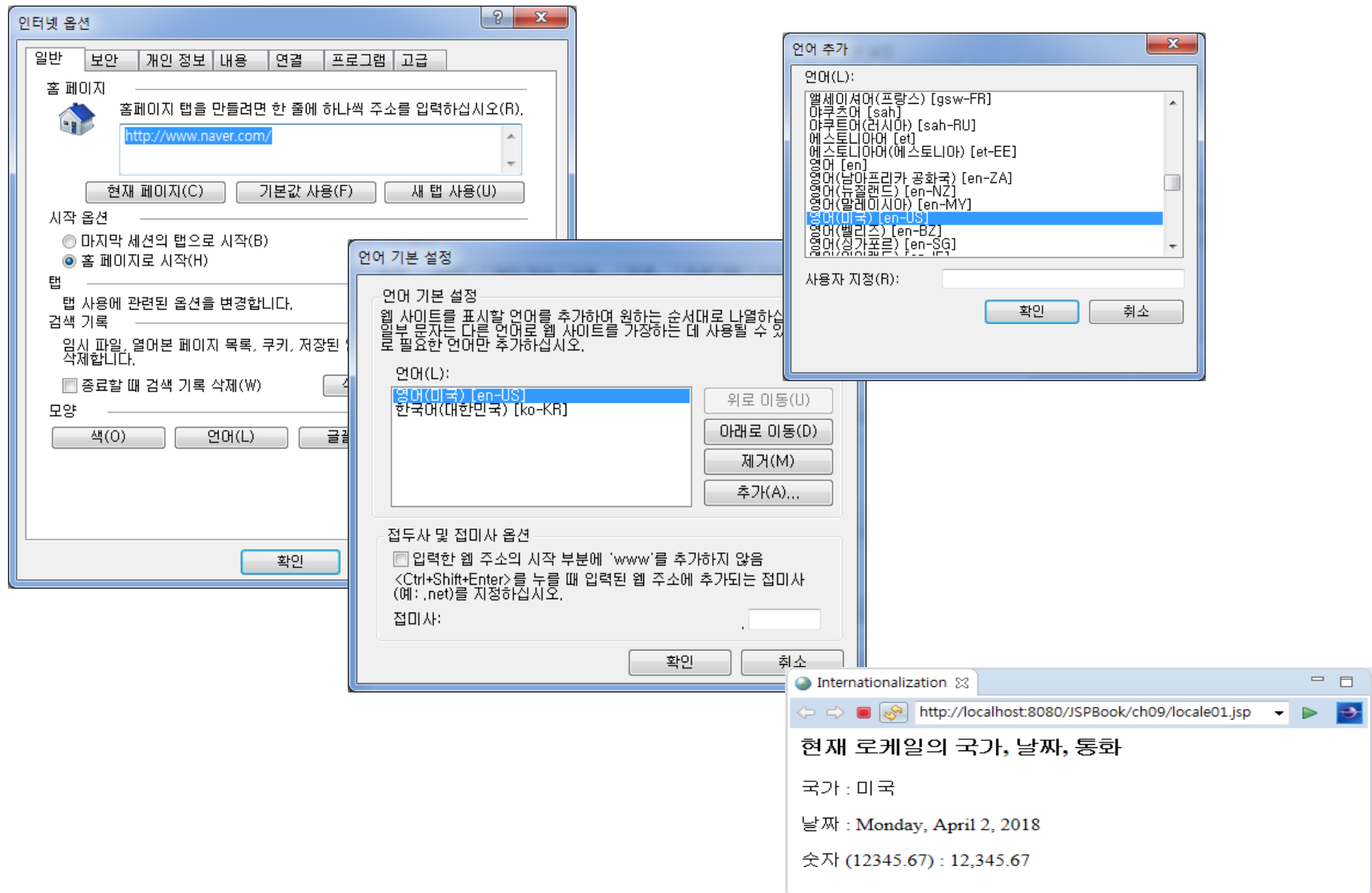
예제 9-1 사용자의 로케일에 따라 국가, 날짜, 통화 형식 출력하기

JSPBook/WebContent/ch09/locale01.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <%page import="java.text.NumberFormat"%>
03 <%page import="java.text.DateFormat"%>
04 <%page import="java.util.*"%>
05 <html>
06 <head>
07 <title>Internationalization</title>
08 </head>
09 <body>
10     <h3>현재 로케일의 국가, 날짜, 통화</h3>
11     <%
12         Locale locale = request.getLocale();
13         Date currentDate = new Date();
14         DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.FULL,
15             locale);
16         NumberFormat numberFormat = NumberFormat.getNumberInstance(locale);
17     %>
18     <p> 국가 : <%=locale.getDisplayCountry()%>
19     <p> 날짜 : <%=dateFormat.format(currentDate)%>
20     <p> 숫자 (12345.67) : <%=numberFormat.format(12345.67)%>
21 </body>
22 </html>
```



2. Locale 클래스를 이용한 다국어 처리



3. JSTL fmt 태그를 이용한 다국어 처리

❖ JSTL fmt 태그

- 다국어 문서 처리를 위한 국제화 및 지역화 태그
- 날짜와 숫자 등을 형식화하는 기능을 제공하는 JSTL 라이브러리인 JSTL fmt 태그는 특정 지역에 따라 다른 메시지를 출력할 때 사용
 - 한글 웹 브라우저는 한글 메시지를 출력할 때, 영문 웹 브라우저는 영어 메시지를 출력할 때 유용

❖ JSTL fmt 태그 라이브러리를 사용 방법

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

- JSTL 라이브러리인 jstl.jar 파일이 필요
 - 배포 사이트: <https://mvnrepository.com/>
 - 다운로드 파일: jstl-1.2.jar

3. JSTL fmt 태그를 이용한 다국어 처리

❖ JSTL fmt 태그의 종류

구분	태그 유형	설명
Locale 설정	setLocale	로케일을 설정합니다.
	requestEncoding	요청 파라미터의 문자 인코딩을 설정합니다.
메시지 처리	bundle	사용할 리소스번들을 설정합니다.
	message	리소스번들에서 로케일에 맞는 메시지를 가져와 출력합니다.
	setBundle	리소스번들을 읽어와 특정 변수에 저장합니다.

구분	태그 유형	설명
날짜	formatDate	날짜 형식을 표현합니다.
	parseDate	문자열에서 원하는 패턴의 날짜 형식으로 변환합니다.
숫자	parseNumber	문자열에서 원하는 패턴의 숫자 형식으로 변환합니다.
	formatNumber	숫자 형식을 표현합니다.
시간	setTimeZone	특정 범위의 시간대를 설정합니다.
	timeZone	시간대를 설정합니다.

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 로케일 설정 태그의 기능과 사용법

- setLocale 태그
 - 국제화 태그가 사용할 로케일을 설정하는 태그

```
<fmt:setLocale  
  value="언어 코드[_국가 코드]"  
  [scope="{page|request|session|application}"]/> //기본 값은 page
```

[setLocale 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>  
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>  
<html>  
<head>  
<title>Internationalization</title>  
</head>  
<body>  
  <%=response.getLocale()%>  
  
  <fmt:setLocale value="ko" />  
  <p><%=response.getLocale()%>  
  
  <fmt:setLocale value="ja" />  
  <p><%=response.getLocale()%>  
  
  <fmt:setLocale value="en" />  
  <p><%=response.getLocale()%>  
</body>  
</html>
```

ko_KR
ko
ja
en

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 로케일 설정 태그의 기능과 사용법

- requestEncoding 태그 : 요청 파라미터의 문자 인코딩을 설정하는 태그

```
<fmt:requestEncoding value="문자 인코딩"/>
```

[requestEncoding 태그 사용 예]

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <fmt:requestEncoding value="euc-kr" />
    요청 파라미터 :
    <%
        out.print(request.getParameter("id"));
    %>
    <form method="post" action="#">
        <p> 아이디 <input type="text" name="id">
            <input type="submit" value="전송">
        </form>
</body>
</html>
```

요청 파라미터 : null

아이디

요청 파라미터 : 홍길순

아이디

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 메시지 처리 태그의 기능과 사용법

■ 리소스번들

- 메시지 처리 태그에서 사용하는 파일로 메시지 번들이라고도 함
- 리소스번들로 사용하는 파일은 보통 WEB-INF/classes/ 폴더에 있음
- 리소스번들은 java.util.Properties 클래스에 정의된 방법으로 메시지를 읽어오기 때문에 확장자가 properties인 파일이 반드시 있어야 함
 - java.util.Properties 클래스는 알파벳이나 숫자, 라틴 문자 외의 언어를 유니코드 값으로 표현

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 메시지 처리 태그의 기능과 사용법

- 리소스번들로 사용하는 *.properties 파일의 종류

*.properties 파일	설명
파일 이름.properties	기본 메시지일 때 사용합니다.
파일 이름_ko.properties	한글 메시지일 때 사용합니다.
파일 이름_en.properties	영어 메시지일 때 사용합니다.

[한글, 영문 리소스번들 작성 예]

```
//message_ko.properties 파일인 경우  
NAME = 홍길순
```

```
//message_en.properties 파일인 경우  
NAME = your name
```

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 메시지 처리 태그의 기능과 사용법

■ bundle 태그

- 사용할 리소스번들을 설정하는 태그로 형식

```
<fmt:bundle basename="리소스번들" [prefix="key 이름"]>  
    //body 내용  
</fmt:bundle>
```

■ message 태그

- bundle 태그에 설정한 리소스번들에서 메시지를 읽어와 출력하는 태그

```
<fmt:message  
    key="메시지 key 이름"  
    [bundle="setBundle 태그의 변수 이름" ]  
    [var="메시지를 저장하는 변수 이름"]  
    [scope="{page|request|session|application}"] /> //기본 값은 page
```

3. JSTL fmt 태그를 이용한 다국어 처리

[bundle 태그와 message 태그 사용 예]

//message_en.properties 파일인 경우

name = admin

hello = How are You

//message.properties 파일인 경우

name = \uAD00\uB9AC\uC790 //"관리자" 유니코드

hello = \uC548\uB155\uD558\uC138\uC694 //"안녕하세요" 유니코드

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <fmt:bundle basename="resourceBundle.message">
    <p> <fmt:message key="name" />
      <fmt:message key="hello" var="msg" />
    <p>${msg}
  </fmt:bundle>
</body>
</html>
```

관리자

안녕하세요

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 메시지 처리 태그의 기능과 사용법

■ setBundle 태그

- 리소스번들을 가져와 변수로 저장한 후 JSP 페이지 어디에서나 사용할 수 있는 태그
- bundle 태그를 대체하여 사용할 수 있음

```
<fmt:setBundle  
    basename="리소스번들"  
    [var="리소스번들을 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/> //기본 값은 page
```

[setBundle 태그와 message 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>  
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>  
<html>  
<head>  
<title>Internationalization</title>  
</head>  
<body>  
    <fmt:setBundle var="message" basename="resourceBundle.message" />  
    <p> <fmt:message bundle="${message}" key="name" />  
        <fmt:message bundle="${message}" key="hello" var="msg" />  
    <p>${msg}  
</body>  
</html>
```

관리자
안녕하세요

3. JSTL fmt 태그를 이용한 다국어 처리

예제 9-2 사용자의 로케일에 따라 리소스번들의 메시지 출력하기

- JSTL fmt 태그를 사용하기 위해 jstl-1.2.jar 파일을 다운로드하여 /WebContent/WEB-INF/lib/ 폴더에 추가
- /src/ 폴더에 다음과 같이 2개의 리소스번들을 작성

JSPBook/src/ch09/com/bundle/myBundle.properties

```
01 title=자바 서버 페이지
02 username=관리자
03 password=1234
```

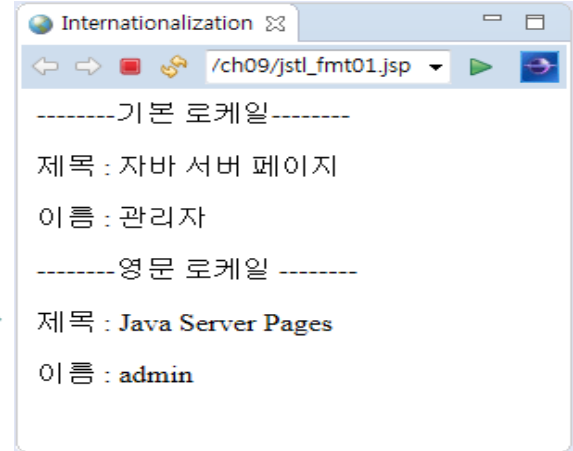
JSPBook/src/ch09/com/bundle/myBundle_en.properties

```
01 title=Java Server Pages
02 username=admin
03 password=1234
```

3. JSTL fmt 태그를 이용한 다국어 처리

JSPBook/WebContent/ch09/jstl_fmt01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
03 <html>
04 <head>
05 <title>Internationalization</title>
06 </head>
07 <body>
08     <p> ——기본 로케일——
09         <fmt:setBundle basename="ch09.com.bundle.
10             myBundle" var="resourceBundle" />
11     <p> 제목 : <fmt:message key="title" bundle=
12         "${resourceBundle}" />
13
14     <p> <fmt:message key="username" var="userMsg" bundle="${resourceBundle}" />
15         이름 : ${userMsg}
16
17     <p> ——영문 로케일 ——
18         <fmt:setLocale value="en" />
19         <fmt:setBundle basename="ch09.com.bundle.myBundle" var=
20             "resourceBundle" />
21     <p> 제목 : <fmt:message key="title" bundle="${resourceBundle}" />
22     <p> 이름 : <fmt:message key="username" bundle="${resourceBundle}" />
23 </body>
24 </html>
```



3. JSTL fmt 태그를 이용한 다국어 처리

❖ 숫자 태그의 기능과 사용법

- formatNumber 태그
 - 숫자를 형식에 맞춰 출력하는 태그

```
<fmt:formatNumber value="형식화할 숫자"  
    [type="{number|currency|percent} "] //기본 값은 number  
    [pattern="사용자 정의 패턴"]  
    [currencyCode="통화 코드"] //type="currency"일 때만 적용  
    [currencySymbol="통화 기호"] //type="currency"일 때만 적용  
    [groupingUsed="천 단위마다 구분 기호{true|false}"] //기본 값은 true  
    [maxIntegerDigits="최대 자릿수"]  
    [minIntegerDigits="최소 자릿수"]  
    [maxFractionDigits="소수점 이하 최대 자릿수"]  
    [minFractionDigits="소수점 이하 최소 자릿수"]  
    [var="형식화된 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/> //기본 값은 page
```

3. JSTL fmt 태그를 이용한 다국어 처리

[formatNumber 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <p>숫자 : <fmt:formatNumber value="1234.567" type="number" />
  <p>통화 : <fmt:formatNumber value="1234.567" type="currency"
currencySymbol="원" />
  <p>퍼센트 : <fmt:formatNumber value="1234.567" type="percent" />
  <p>패턴(.0000): <fmt:formatNumber value="1234.567" pattern=".0000" />
</body>
</html>
```

숫자 : 1,234.567

통화 : 원 1,235

퍼센트 : 123,457%

패턴(.0000): 1234.5670

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 숫자 태그의 기능과 사용법

- parseNumber 태그
 - formatNumber 태그와 반대로 사용자가 설정한 패턴 문자열에서 숫자를 추출하는 태그

```
<fmt:parseNumber value="파싱할 숫자"  
    [type="{number|currency|percent}"]  
    [pattern="사용자 정의 패턴"]  
    [parseLocale="파싱의 기본 형식 패턴을 제공하는 로케일"]  
    [integerOnly="{true|false}"]  
    [var="파싱 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/>
```

3. JSTL fmt 태그를 이용한 다국어 처리

[parseNumber 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <p>패턴(없음) : <fmt:parseNumber value="1234.567" />
  <p>패턴(0000.000) : <fmt:parseNumber value="1234.567" pattern="0000.000" />
  <p>패턴(####.###) : <fmt:parseNumber value="1234.567" pattern="####.###" />
</body>
</html>
```

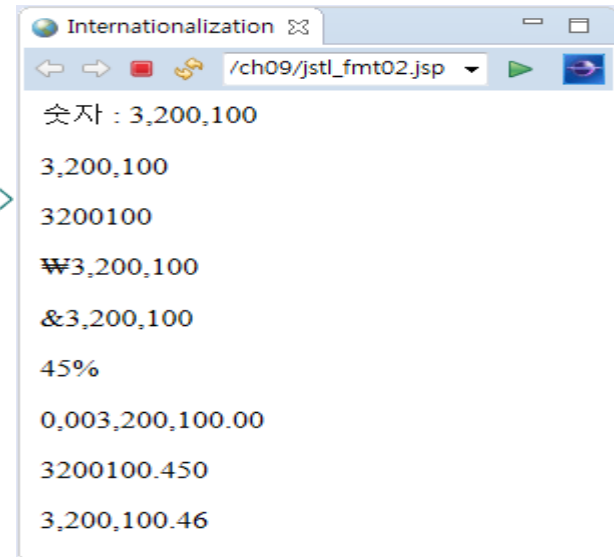
패턴(없음) : 1234.567
패턴(0000.000) : 1234.567
패턴(####.###) : 1234.567

3. JSTL fmt 태그를 이용한 다국어 처리

예제 9-3 사용자의 로케일에 따라 숫자를 다양한 형식에 맞춰 출력하기

JSPBook/WebContent/ch09/jstl_fmt02.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
03 <html>
04 <head>
05 <title>Internationalization</title>
06 </head>
07 <body>
08     <p> 숫자 : <fmt:formatNumber value="3200100" />
09     <p> <fmt:formatNumber value="3200100"
10         type="number" />
11     <p> <fmt:formatNumber value="3200100"
12         type="number" groupingUsed="false" />
13     <p> <fmt:formatNumber value="3200100"
14         type="currency" groupingUsed="true" />
15     <p> <fmt:formatNumber value="3200100" type="
16         "currency" currencySymbol="&" />
17     <p> <fmt:formatNumber value="0.45" type="percent" />
18     <p> <fmt:formatNumber value="3200100" minIntegerDigits="10"
19         minFractionDigits="2" />
20     <p> <fmt:formatNumber value="3200100.45" pattern=".000" />
21     <p> <fmt:formatNumber value="3200100.456" pattern="#,#00.0#" />
22 </body>
23 </html>
```



3. JSTL fmt 태그를 이용한 다국어 처리

❖ 날짜 태그의 기능과 사용법

- formatDate 태그
 - 날짜 정보를 담고 있는 객체를 형식화하여 출력하는 태그

```
<fmt:formatDate value="형식화할 날짜"  
    [type="{time|date|both}"]  
    [dateStyle="{default|short|medium|long|full}"]  
    [timeStyle="{default|short|medium|long|full}"]  
    [pattern="사용자 정의 패턴"]  
    [timeZone="타임존"]  
    [var="형식화된 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/>
```

3. JSTL fmt 태그를 이용한 다국어 처리

[formatDate 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <jsp:useBean id="now" class="java.util.Date" />
    <p> date 형식 : <fmt:formatDate value="${now}" type="date" />
    <p> time 형식 : <fmt:formatDate value="${now}" type="time" />
    <p> both 형식 : <fmt:formatDate value="${now}" type="both" />

</body>
</html>
```

date 형식 : 2018. 9. 5.

time 형식 : 오후 5:02:59

both 형식 : 2018. 9. 5. 오후 5:02:59

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 날짜 태그의 기능과 사용법

- parseDate 태그
- 문자열로 표시된 날짜와 시간 값을 java.util.Date로 변환하는 태그

```
<fmt:parseDate value="파싱할 날짜"  
    [type="{time|date|both}"]  
    [dateStyle="{default|short|medium|long|full}"]  
    [timeStyle="{default|short|medium|long|full}"]  
    [pattern="사용자 정의 패턴"]  
    [timeZone="타임존"]  
    [parseLocale="파싱의 기본 형식 패턴을 제공하는 로케일"]  
    [var="파싱 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/>
```


3. JSTL fmt 태그를 이용한 다국어 처리

[parseDate 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <p><fmt:parseDate value="20180402230605" pattern="yyyyMMddHHmmss" />
  <p> <fmt:parseDate pattern="yyyyMMddHHmmss"> 20180402230605 </fmt:parseDate>
  <p><fmt:parseDate value="20180402230605" pattern="yyyyMMddHHmmss"
var="date" scope="page" />
  <p><fmt:formatDate value="${date}" pattern="yyyy-MM-dd HH:mm" />
</body>
</html>
```

Mon Apr 02 23:06:05 KST 2018

Mon Apr 02 23:06:05 KST 2018

2018-04-02 23:06

3. JSTL fmt 태그를 이용한 다국어 처리

예제 9-4 사용자의 로케일에 따라 날짜 형태 출력하기

JSPBook/WebContent/ch09/jstl_fmt03.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
03 <html>
04 <head>
05 <title>Internationalization</title>
06 </head>
07 <body>
08     <p> <jsp:useBean id="now" class="java.
        util.Date" />
09     <p> <fmt:formatDate value="${now}"
        type="date" />
10     <p> <fmt:formatDate value="${now}" type="time" />
11     <p> <fmt:formatDate value="${now}" type="both" />
12     <p> <fmt:formatDate value="${now}" type="both" dateStyle="default"
        timeStyle="default" />
13     <p> <fmt:formatDate value="${now}" type="both" dateStyle="short"
        timeStyle="short" />
14     <p> <fmt:formatDate value="${now}" type="both" dateStyle="medium"
        timeStyle="medium" />
15     <p> <fmt:formatDate value="${now}" type="both" dateStyle="long"
        timeStyle="long" />
16     <p> <fmt:formatDate value="${now}" type="both" dateStyle="full"
        timeStyle="full" />
17     <p> <fmt:formatDate value="${now}" type="both" pattern="yyyy년MM월dd일 HH시
        mm분ss초 E요일" />
18 </body>
19 </html>
```



3. JSTL fmt 태그를 이용한 다국어 처리

❖ 시간 태그의 기능과 사용법

- timeZone 태그
 - 시간대별로 시간을 처리하는 태그

```
<fmt:timeZone value="타임존">
    //
</fmt:timeZone>
```

[timeZone 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization
</title>
</head>
<body>
    <fmt:timeZone value="KST" />
    <p><fmt:timeZone value="GMT">
        런던 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"
            timeStyle="full" />
    </fmt:timeZone>
    </p>
</body>
</html>
```

한국 : 2018년 9월 5일 수요일 오후 5시 4분 23초 대한민국 표준시
뉴욕 : 2018년 9월 5일 수요일 오전 12시 4분 23초 GMT-08:00
런던 : 2018년 9월 5일 수요일 오전 8시 4분 23초 그리니치 표준시

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 시간 태그의 기능과 사용법

- setTimeZone 태그
 - 특정 영역 범위의 시간대별로 시간을 처리하는 태그

```
<fmt:setTimeZone value="타임존"  
    [var="시간대 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/>
```

3. JSTL fmt 태그를 이용한 다국어 처리

[setTimeZone 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <jsp:useBean id="now" class="java.util.Date" />
    <p> 한국 : <fmt:formatDate value="${now}" type="both" dateStyle="full"
        timeStyle="full"/>
    <p><fmt:setTimeZone value="GMT-8"/>
        뉴욕 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"
            timeStyle="full" />
    <p><fmt:setTimeZone value="GMT"/>
        런던 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"
            timeStyle="full" />
</body>
</html>
```

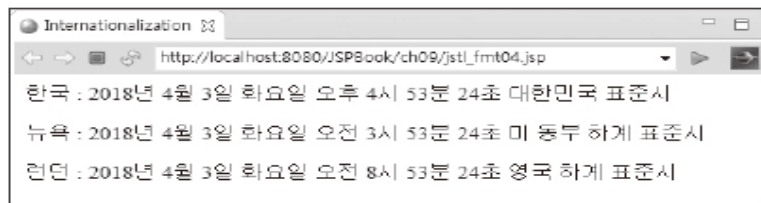
한국 : 2018년 9월 5일 수요일 오후 5시 5분 10초 대한민국 표준시
뉴욕 : 2018년 9월 5일 수요일 오전 12시 5분 10초 GMT-08:00
런던 : 2018년 9월 5일 수요일 오전 8시 5분 10초 그리니치 표준시

3. JSTL fmt 태그를 이용한 다국어 처리

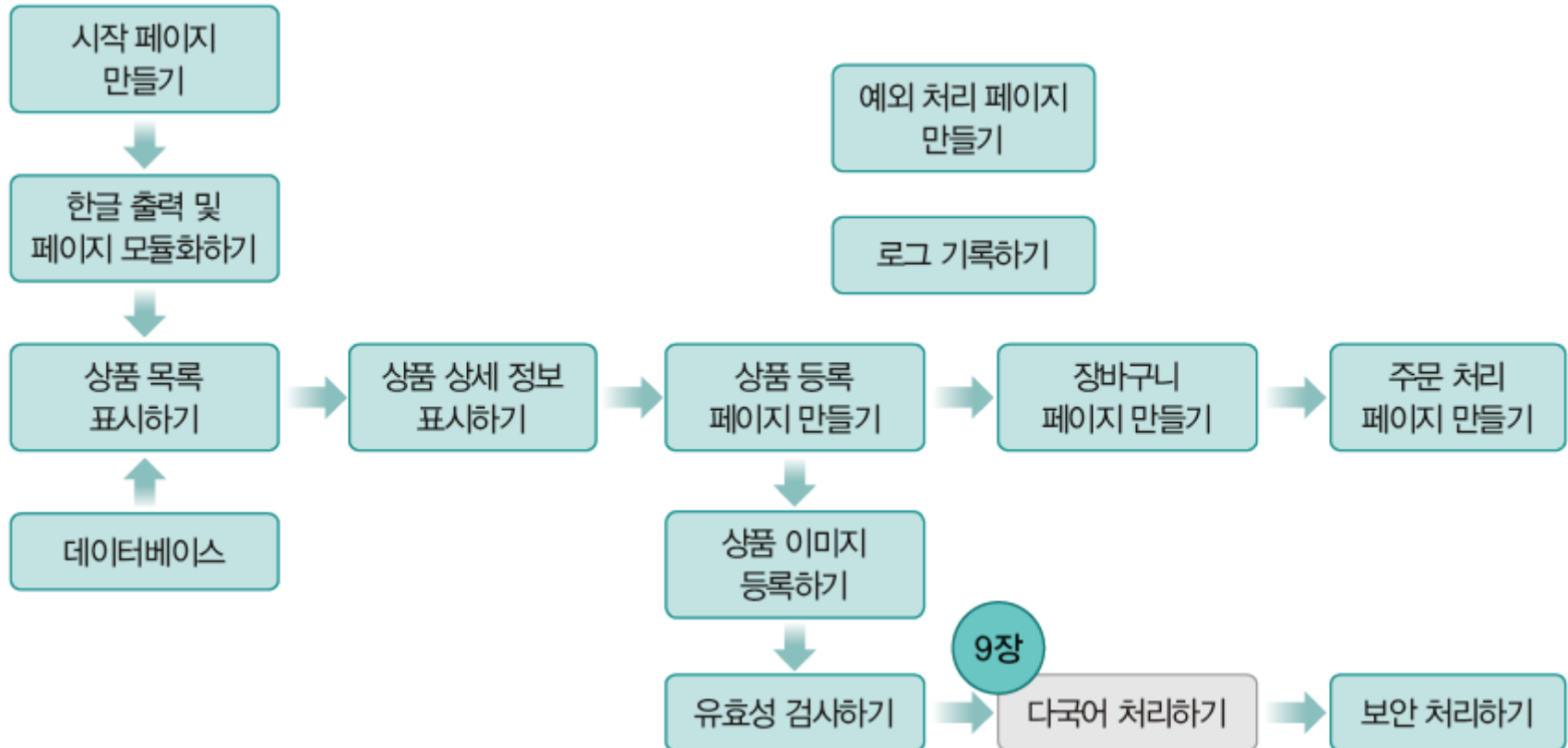
예제 9-5 사용자의 로케일에 따라 타임존 출력하기

JSPBook/WebContent/ch09/jstl_fmt04.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
03 <html>
04 <head>
05 <title>Internationalization</title>
06 </head>
07 <body>
08     <jsp:useBean id="now" class="java.util.Date" />
09     <p> 한국 : <fmt:formatDate value="${now}" type="both" dateStyle="full"
10         timeStyle="full" />
11     <p> <fmt:timeZone value="America/New_York">
12         뉴욕 : <fmt:formatDate value="${ now }" type="both"
13             dateStyle="full" timeStyle="full" />
14     </fmt:timeZone>
15     <p> <fmt:timeZone value="Europe/London">
16         런던 : <fmt:formatDate value="${ now }" type="both"
17             dateStyle="full" timeStyle="full" />
18     </fmt:timeZone>
19 </body>
20 </html>
```



4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기



4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

상품 등록

http://localhost:8080/WebMarket/addProduct.jsp?language=ko

Home

상품 등록

상품 코드

상품명

가격

상세 설명

제조사

분류

재고수

상태 ☐ 신규제품 ☐ 중고제품 ☐ 재생제품

이미지

상품 등록

http://localhost:8080/WebMarket/addProduct.jsp?language=en

Home

Product Addition

[Korean](#) [English](#)

Product ID

Name

Unit Price

Description

Manufacturer

Category

Units in Stock

Condition ☐ New ☐ Old ☐ Refurbished

Image

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

예제 9-6 JSTL fmt 태그를 이용하여 상품 등록 페이지의 다국어 처리하기

- JSTL 라이브러리 jstl-1.2.jar 등록하기
- 한글 리소스번들 만들기

WebMarket/src/bundle/message.properties

```
01 title = 상품 등록
02 productId = 상품 코드
03 pname = 상품명
04 unitPrice = 가격
05 description = 상세 설명
06 manufacturer = 제조사
07 category = 분류
08 unitsInStock = 재고 수
09 condition = 상태
10 productImage = 이미지
11 condition_New = 신규 제품
12 condition_Old = 중고 제품
13 condition_Refurbished = 재생 제품
14 button = 등록
```

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

■ 영문 리소스번들 만들기

WebMarket/src/bundle/message_en.properties

```
01 title = Product Addition
02 productId = Product ID
03 pname = Name
04 unitPrice =Unit Price
05 description = Description
06 manufacturer = Manufacturer
07 category = Category
08 unitsInStock = Units in Stock
09 condition = Condition
10 productImage = Image
11 condition_New = New
12 condition_Old = Old
13 condition_Refurbished = Refurbished
14 button = Insert
```

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

■ 상품 등록 페이지의 다국어 처리하기

WebMarket/WebContent/addProduct.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
03 ... (생략) ...
04 <fmt:setLocale value='${%= request.getParameter("language") %}' />
05 <fmt:bundle basename="bundle.message" >
06 <jsp:include page="menu.jsp" />
07 <div class="jumbotron">
08     <div class="container">
09         <h1 class="display-3"><fmt:message key="title" /></h1>
10     </div>
11 </div>
12 <div class="container">
13     <div class="text-right">
14         <a href="?language=ko" >Korean</a>|<a href="?language=en" >
            English</a>
15     </div>
16     <form name="newProduct" action="./processAddProduct.jsp"
17         class="form-horizontal" method="post" enctype="multipart/form-
            data">
18         <div class="form-group row">
19             <label class="col-sm-2"><fmt:message key="productId" />
                </label>
```

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

```
20         <div class="col-sm-3">
21             <input type="text" id="productId" name="productId"
22                 class="form-control">
23         </div>
24     </div>
25     <div class="form-group row">
26         <label class="col-sm-2"><fmt:message key="pname" /></label>
27         <div class="col-sm-3">
28             <input type="text" id="name" name="name" class="form-
29                 control">
30         </div>
31     </div>
32     <div class="form-group row">
33         <label class="col-sm-2"><fmt:message key="unitPrice" />
34         </label>
35         <div class="col-sm-3">
36             <input type="text" id="unitPrice" name="unitPrice"
37                 class="form-control">
38         </div>
39     </div>
40     <div class="form-group row">
41         <label class="col-sm-2"><fmt:message key="description" />
42         </label>
43         <div class="col-sm-5">
```

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

```
39         <textarea name="description" cols="50" rows="2"
40             class="form-control">/textarea>
41     </div>
42 </div>
43 <div class="form-group row">
44     <label class="col-sm-2"><fmt:message key="manufacturer" />
45     </label>
46     <div class="col-sm-3">
47         <input type="text" name="manufacturer" class="form-
48             control">
49     </div>
50 </div>
51 <div class="form-group row">
52     <label class="col-sm-2"><fmt:message key="category" />
53     </label>
54     <div class="col-sm-3">
55         <input type="text" name="category" class="form-
56             control">
57     </div>
58 </div>
59 <div class="form-group row">
60     <label class="col-sm-2"><fmt:message key="unitsInStock" />
61     </label>
62     <div class="col-sm-3">
63         <input type="text" id="unitsInStock" name=
64             "unitsInStock" class="form-control">
65     </div>
66 </div>
```

4. [웹 쇼핑몰] 상품 등록 페이지의 다국어 처리하기

```
61         <div class="form-group row">
62             <label class="col-sm-2"><fmt:message key="condition" />
63             </label>
64             <div class="col-sm-5">
65                 <input type="radio" name="condition" value="New ">
66                 <fmt:message key="condition_New" />
67                 <input type="radio" name="condition" value="Old">
68                 <fmt:message key="condition_Old" />
69                 <input type="radio" name="condition" value=
70                 "Refurbished"><fmt:message key="condition_Refurbished" />
71             </div>
72         </div>
73     </div>
74     <div class="form-group row">
75         <label class="col-sm-2"><fmt:message key="productImage" />
76         </label>
77         <div class="col-sm-5">
78             <input type="file" name="productImage" class="form-
79             control">
80         </div>
81     </div>
82     <div class="form-group row">
83         <div class="col-sm-offset-2 col-sm-10 ">
84             <input type="button" class="btn btn-primary"
85             value="<fmt:message key="button" />"
86             onclick="CheckAddProduct()">
87         </div>
88     </div>
89 </form>
90 </fmt:bundle>
91 </body>
92 </html>
```

