

- ▶ EL(Expression Language)
 - ▶ 표현식 언어 또는 익스프레션 언어
 - ▶ JSTL 1.0에 소개, JSP 2.0/JSTL 1.1에 추가된 스크립트 언어
 - ▶ 아파치 탐켓 5.0부터 사용 가능
 - ▶ JSP 페이지에서 자바코드 대신 액션태그 엘리먼트의 속성에 값을 지정하는 역할
 - ▶ 표현식(<%= %>) 대용 효과
- ▶ 주요기능
 - ▶ 리터럴 데이터 출력
 - ▶ 다양한 연산자와 연산결과 출력 지원
 - ▶ 4개의 scope[page, request, session, application) 속성 값 출력
 - ▶ JSTL과 연동
- ▶ JSP 모델 1 소스 코드의 단점
 - ▶ HTML, 스크립트릿, 표현식(<%= %> 중첩으로 소스 프로그램 가독성 저하
 - ▶ 디버깅 어려움

<%= %> 대용

\${ }

복 습 하 기

▶ EL의 연산자

▶ 종류

- ▶ 산술연산자
- ▶ 비교연산자
- ▶ 논리연산자
- ▶ empty 연산자
- ▶ 기타



`${ }`

복습하기

▶ EL 내장객체 [EL의 식에서만 사용]

`${ }`

구분	내장객체	설명
Scope	pageScope	page 영역에 존재하는 객체의 참조
	requestScope	request 영역에 존재하는 객체의 참조
	sessionScope	session 영역에 존재하는 객체의 참조
	applicationScore	application 영역에 존재하는 객체의 참조
요청 파라메타	param	요청 파라메타 값을 단일 값으로 반환
	paramValues	요청 파라메타의 값을 배열로 반환
헤더 값	header	요청 헤더명의 정보를 단일 값으로 반환
	headerValues	요청 헤더명의 정보를 배열로 반환
쿠키 값	Cookies	쿠키명의 값을 반환
JSP	pageContext	PageContext 객체를 참조할 때
초기 파라메타	initParam	컨텍스트의 초기화 매개변수명의 값을 반환

▶ 요청 파라메타의 표기

▶ 1) param

- ▶ 웹 브라우저의 입력 폼에서 전송된 단일 값을 얻을 때 사용하는 내장객체
- ▶ JSP 내장객체의 `request.getParameter()` 메서드와 동일

표기법	<code>\${param.필드명}</code> 또는 <code>\${param["필드명"]}</code>
-----	---

▶ 2) paramValue

- ▶ 웹 브라우저의 입력 폼에서 check 태그나 select 태그로 전송된 배열 값을 얻을 때 사용하는 내장객체
- ▶ 인덱스는 0부터 시작
- ▶ JSP 내장객체의 `request.getParameterValues()` 메서드와 동일

표기법	<code>\${paramValue.배열명[인덱스]}</code> 또는 <code>\${param["배열명"][인덱스]}</code>
-----	--

Part2. 기초 프로그래밍

- ▶ Chapter4. JSP 기본문법
- ▶ Chapter5. JSP 내장객체
- ▶ Chapter6. JSP 입력 폼 설계
- ▶ Chapter7. JSP와 DB 연동
- ▶ Chapter8. 자바빈과 액션태그
- ▶ Chapter9. 쿠키와 세션
- ▶ Chapter10. 서블릿
- ▶ Chapter11. DBCP
- ▶ Chapter12. EL
- ▶ Chapter13. JSTL

문법과
기초 프로그래밍 실습

제13장 JSTL



**Introducing The
JSP Standard
Tag Library**

1. JSTL?
2. core 라이브러리
3. format 라이브러리
4. sql 라이브러리
5. xml 라이브러리
6. functions 라이브러리



1. JSTL?

2. core 라이브러리

3. format 라이브러리

4. sql 라이브러리

5. xml 라이브러리

6. functions 라이브러리

1. JSTL?
2. core 라이브러리
3. format 라이브러리
4. sql 라이브러리
5. xml 라이브러리
6. functions 라이브러리



13. JSTL

1. JSTL?

- 2. core 라이브러리
- 3. format 라이브러리
- 4. sql 라이브러리
- 5. xml 라이브러리
- 6. functions 라이브러리



1. JSTL (JSP Standard Tag Library)

▶ JSTL?

- ▶ JSP Standard Tag Library) 약어. JSP 표준 태그 라이브러리
- ▶ 라이브러리란?
 - ▶ 여러 프로그램이 공통으로 사용하는 코드의 코드 집합
- ▶ core, format, database, xml, functions 5개의 라이브러리
- ▶ functions 은 함수, 4 개 라이브러리들은 커스텀 액션 태그

라이브러리	기 능 설 명
core	● 변수 선언. 흐름의 제어, 다른 JSP 페이지로 제어 이동 기능
format	● 숫자, 날짜, 시간을 형식 지정과 국제화, 다국어 지원 기능
database	● 데이터베이스의 데이터 입력, 수정, 삭제, 조회 기능
xml	● xml 문서 처리 기능
functions	● 문자열 처리 함수 기능

1.1 JSTL 지시어 표기법

- ▶ 소스 코드 상단에 taglib 지시어 표기

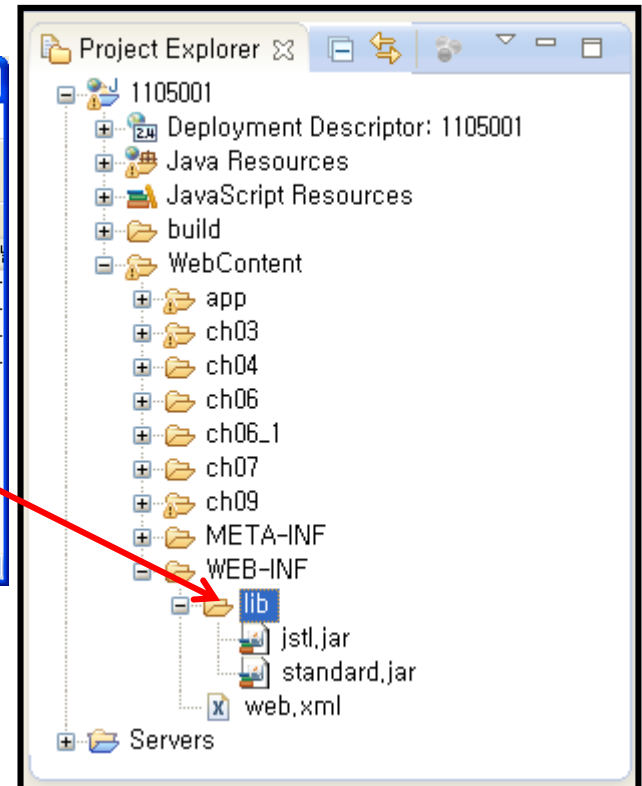
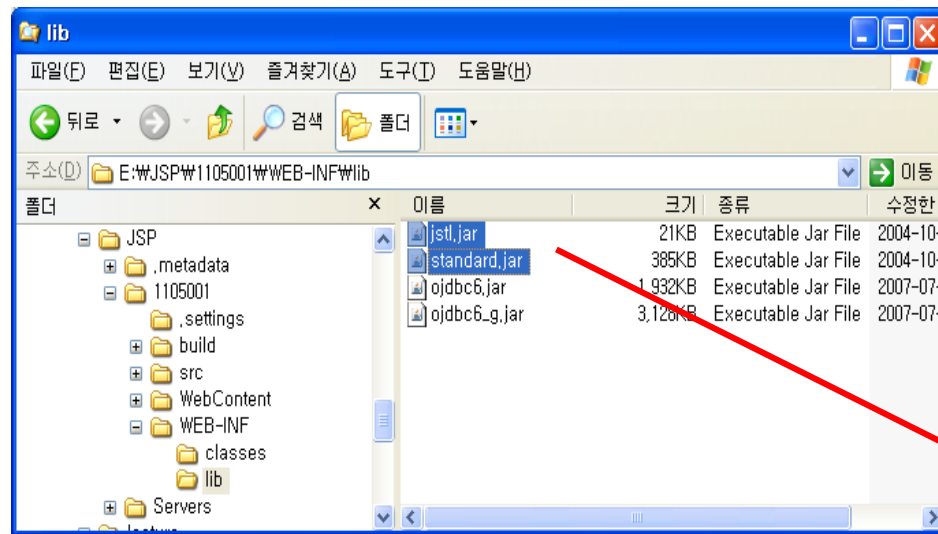
태그	<code><%@ taglib prefix="접두어" uri="uri" %></code>
----	---

라이브러리	JSTL 접두어와 지시어
core	<code><%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %></code>
format	<code><%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %></code>
database	<code><%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %></code>
xml	<code><%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %></code>
functions	<code><%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %></code>

JSP 페이지 상단에 코딩

- ▶ **JSTL 1.1** 라이브러리를 구성하는 파일
 - ▶ `jstl.jar`
 - ▶ `standard.jar`
- ▶ 다운로드 사이트
 - ▶ "<http://tomcat.apache.org>"
 - ▶ "[jakarta-taglibs-standard-1.1.2.zip](#)" 파일 다운로드
 - ▶ 압축을 푼다.
- ▶ 저장 위치
 - ▶ `WEB-INF/lib`
 - ▶ `Jstl.jar`, `standard.jar` 파일 복사

- ▶ [Y class] [강의자료실] 다운로드
 - ▶ jakarta-taglibs-standard-1.1.2.zip
 - ▶ 압축 풀고, 2개 파일 복사 [WEB-INF\LIB]



2. core 라이브러리

▶ 지시어

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

▶ 태그 목록

기능	태그	설명
표현언어 지원	<code><c:set/></code>	변수 설정
	<code><c:remove/></code>	변수값 제거
	<code><c:out/></code>	표현식 대체, 웹브라우저로 출력
	<code><c:catch/></code>	예외처리 사용
흐름제어	<code><c:if/></code>	조건 처리
	<code><c:choose/></code>	여러 조건에 따른 처리
	<code><c:forEach/></code>	반복처리
	<code><c:forTokens/></code>	구분자로 분리된 각 토큰을 처리
URL 관리	<code><c:import/></code>	외부자원 포함
	<code><c:redirect/></code>	지정 경로 이동
	<code><c:url/></code>	url 재작성

2.1 <c:set> 태그

- ▶ <c:set>
 - ▶ 변수와 초기값 지정
 - ▶ value에 문자열이나 EL식 사용
 - ▶ scope의 기본값은 page
 - ▶ 변수는 EL식에서만 사용
 - ▶ JSP의 **setAttribute()**와 같은 역할

page | request |
session | application

태그 1	<c:set var="변수명" value="값" [scope="속성"] />
태그 2	<c:set var="변수명" [scope="속성"] >body contents </c:set>
태그 3	<c:set value="값" target="target" property="프로퍼티명" />
태그 4	<c:set target="target" property="프로퍼티명" > body contents </c:set>

2.1 <c:set> 태그

속성	설 명
var	변수명
value	초기값
scope	page, request, session, application 영역, page가 기본값
target	target 지정
property	프로퍼티명

EL식으로

예

- `<c:set var="msg" value="${'hello'}" />`
- `<c:set var="age" scope="page">${30}</c:set>`

예

- `<% request.setAttribute("key", "값" %>`
`=> <c:set var="key" value="${값}" scope="request" />`

예

- `<% session.setAttribute("key", "값" %>`
`=> <c:set var="key" value="${값}" scope="session" />`

2.2 <c:out> 태그

▶ <c:out>

- ▶ 웹브라우저에 값 출력
- ▶ value 값은 문자열이나 EL식 사용
- ▶ scope의 기본값은 page

Value의 값이 널이면 기본값 출력

태그 1 `<c:out value="값" escapeXml={true|false} [default="기본값"] />`

태그 2 `<c:out value="값" escapeXml={true|false} > 기본값 </c:out>`

EL식으로

예

- `<c:out value="${login}" default="Guest" />`
- `<c:out value="${age}" > ${30} </c:out>`

`<%= %>` 또는 `out.println()` 대용

2.3 <c:remove> 태그

- ▶ <c:remove>
 - ▶ 변수 값 제거
 - ▶ scope의 기본값은 page
 - ▶ JSP의 removeAttribute()와 같은 역할

태그	<code><c:remove var="변수명" [scope="속성"] /></code>
----	--

예	<ul style="list-style-type: none">● <code><c:remove var="age" scope="page" /></code>
---	--

예	<ul style="list-style-type: none">● <code><% session.removeAttribute("key"); %></code> => <code><c:remove var="key" scope="session" /></code>
---	--

2.4 <c:catch>

- ▶ <c:catch>
 - ▶ JAVA의 try ~ catch~ 대체 태그
 - ▶ 결과 변수에 저장
 - ▶ 예외 발생시 메시지를 변수에 저장

정상처리시 결과 저장

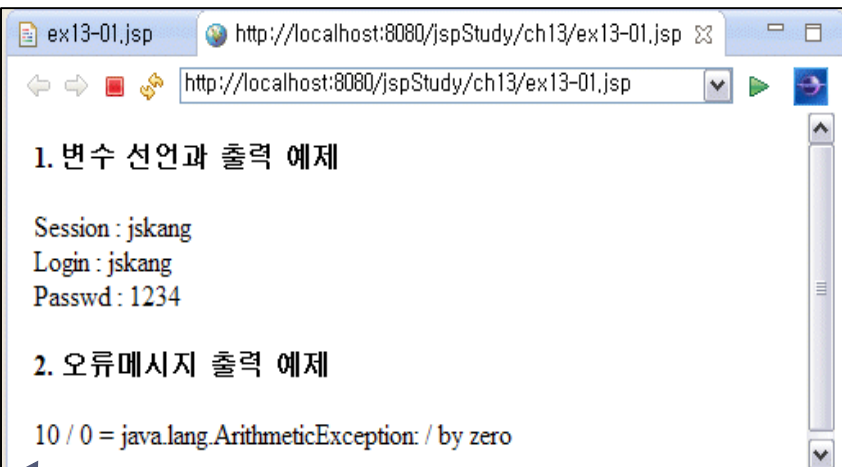
태그	<c:catch [var="변수명"]> actions </c:catch>
----	--

예

```
• <c:catch var="errmsg"> <%= 1/0 %> </c:catch>  
  <c:out value="${errmsg}" />
```

【예제 13.1】

login, passwd 변수의 초기값으로 "jskang", "1234"으로 "session"으로 선언하고, login의 session 값, login, passwd 변수를 출력하고, 세션 영역의 login의 변수 값을 제거하시오. 그리고 10/0의 처리에 대한 오류 메시지를 출력하는 프로그램을 작성하시오.



10 / 2 = 2 추가

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4
5 <c:set var="login" value="jskang" scope="session" />
6 <c:set var="passwd" value="1234" />
7
8 <h4>1. 변수 선언과 출력 예제</h4>
9 Session : <%= session.getAttribute("login") %> <br>
10 Login   : <c:out value="${login}" /> <br>
11 Passwd  : <c:out value="${passwd}" />
12 <c:remove var="login" scope="session" />
13 <h4>2. 오류메시지 출력 예제 </h4>
14 10 / 0 =
15 <c:catch var="errmsg"> <%= 10 / 0 %> </c:catch>
16 <c:out value="${errmsg}" />
```

`${sessionScope.login}`

JSTL core의 흐름 제어

- ▶ JSTL core의 흐름 제어 태그
 - ▶ `<c:if>`
 - ▶ `if` 문과 동일. `if`만 가능
 - ▶ `<c:choose>`
 - ▶ `if ~ else` 문. `<c:if>` 대체 명령문
 - ▶ `<c:forEach>`
 - ▶ 배열, 컬렉션, 맵에 저장된 값들의 순차적 처리
 - ▶ `<c:forTokens>`
 - ▶ 구분자(delims)에 맞추어 반복 처리

2.5 <c:if> 태그

- ▶ <c:if>
 - ▶ If문과 동일하나 if만 가능
 - ▶ 조건에 따라 true 혹은 false가 변수명에 저장
 - ▶ scope의 기본값은 page

태그1 `<c:if test="조건" var="변수명" [scope="{page|request|session|application}"]/>`

태그2 `<c:if test="조건" [var="변수명"] [scope="속성"]/>`
`body content`
`</c:if>`

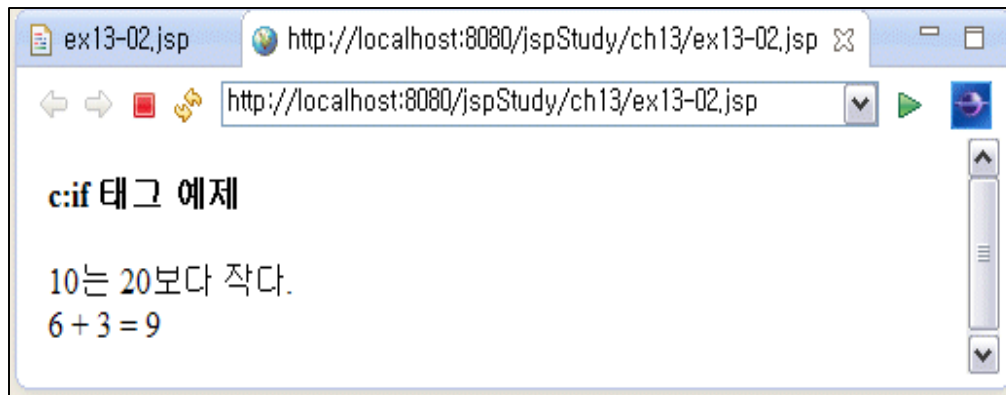
EL식으로

예

- `<c:if test="${param.login == 'admin'}"> 관리자입니다. </c:if>`
- `<c:if test="${6+3 == 9}"> 6 + 3 = 9 </c:if>`

【예제 13.2】 10과 20을 비교하여 참일 때 "10은 20 보다 작다.", "6+3=9" 등식이 참일 때 "6+3=9"를 출력하는 프로그램을 작성하시오.

생략



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <h4> c:if 태그 예제 </h4>
5 <c:if test="${10 < 20}"> 10는 20보다 작다. </c:if><br>
6 <c:if test="${6+3 == 9}"> 6 + 3 = 9 </c:if>
```

2.6 <c:choose> <c:when><c:otherwise>

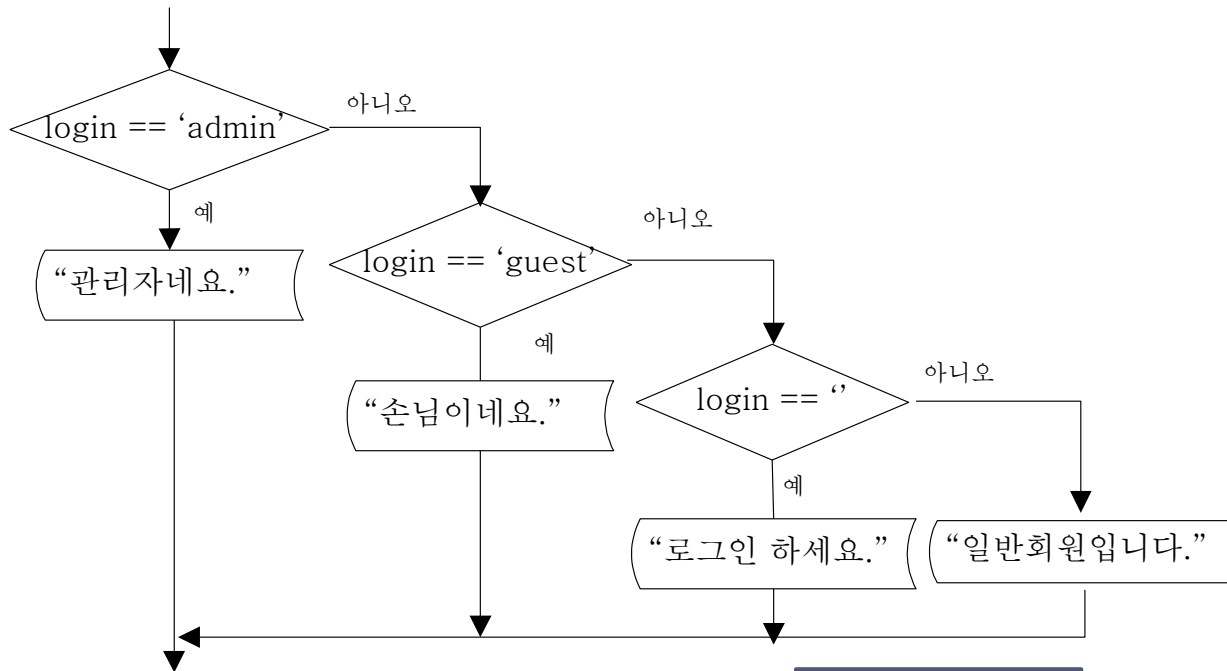
- ▶ <c:choose> <c:when> <c:otherwise>
 - ▶ <c:choose> : switch문과 동일
 - ▶ if ~ else 문의 <c:if> 대체 명령문
 - ▶ scope의 기본값은 page

태그	<pre><c:choose> body content { <c:when test="{조건1}"> 처리내용-1 </c:when> 서브태그 ... <c:otherwise> 처리내용-n</otherwise> 서브태그 } </c:choose></pre>
----	--

서브태그 1	<pre><c:when test="조건1"> body content </c:when></pre>
--------	---

서브태그 2	<pre><c:otherwise> block-n </c:otherwise></pre>
--------	---

2.6 <c:choose> <c:when><c:otherwise>



EL 식으로

예

- <c:choose>

```
<c:when test="${param.login == 'admin'}">관리자네요.</c:when>
```

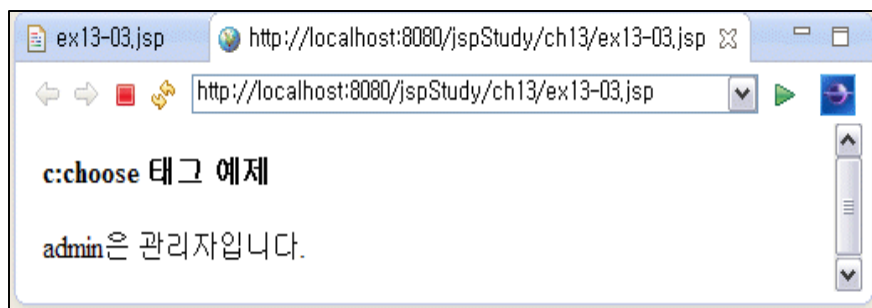
```
<c:when test="${param.login == 'guest'}">손님이네요.</c:when>
```

```
<c:when test="${empty param.login}">로그인하세요.</c:when>
```

```
<c:otherwise>일반회원입니다.</c:otherwise>
```

```
</c:choose>
```

【예제 13.3】 login 값이 관리자("admin"), 손님("guest") 또는 일반회원(null)인지 출력하는 프로그램을 작성하고, login 변수가 "admin"일 때 처리하시오.



```
ex13-03.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
5
6 <h4> c:choose 태그 예제 </h4>
7 <c:set var = "login" value="admin" />
8 ${login}은
9 <c:choose>
10   <c:when test="${login == 'admin'}">관리자 입니다.</c:when>
11   <c:when test="${login == 'guest'}">손님 입니다.</c:when>
12   <c:when test="${empty login}">로그인하세요.</c:when>
13 </c:choose>
14
```

login의 값을 변경하여 실행.

2.7 <c:forEach> 태그

▶ <c:forEach>

- ▶ 배열, 컬렉션, 맵에 저장된 값들의 순차적 처리
- ▶ 객체의 길이만큼 반복 가능
 - ▶ Java의 for문과 동일

배열, collection, Map 등

태그1

```
<c:forEach var="변수명" items="반복할 객체명" varStatus="변수상태명"
           begin="시작값" end="최종값" [step="단계값"]>
```

body content

```
</c:forEach>
```

태그2

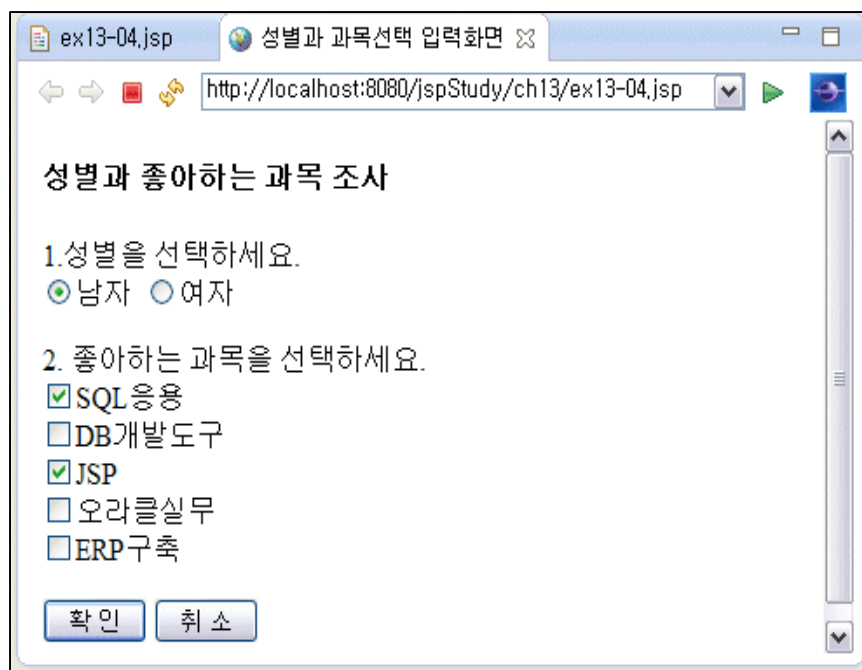
```
<c:forEach [var="변수명"] begin="시작값" end="최종값" [step="단계값"]>
```

body content

```
</c:forEach>
```

속성	동적값	타입	설 명
var	false	String	아이템(items)을 갖고 있는 변수
items	True	any	배열, 컬렉션(collection), Map 등
varStatus	false	String	반복 상태 값을 갖고 있는 변수
begin	true	int	시작값
end	true	int	최종값
step	true	int	단계값

【예제 13.4】 성별과 좋아하는 과목을 조사하는 화면을 만들고, 전송된 값을 출력하는 프로그램을 작성하시오. (예제 6.3과 예제 6.4와 동일한 예제로 "ex13-04-1.jsp" 소스 코드의 가독성이 좋아진다.)



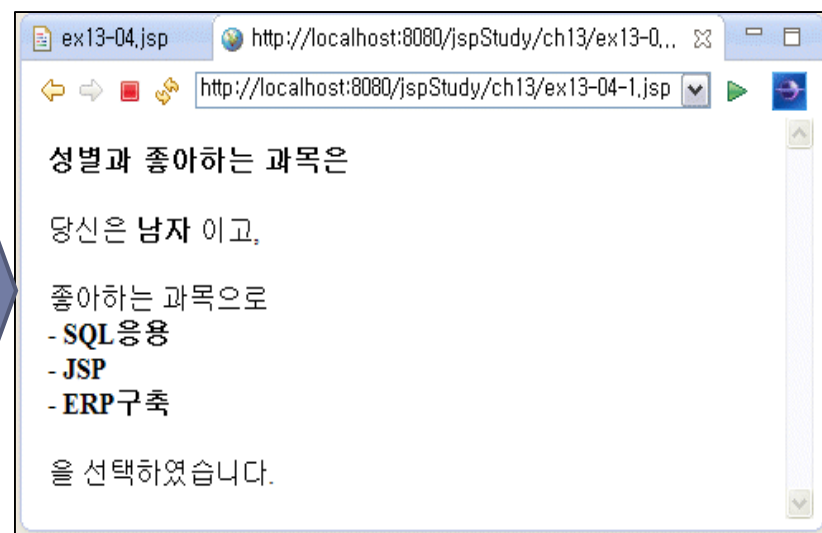
ex13-04.jsp 성별과 과목선택 입력화면

http://localhost:8080/jspStudy/ch13/ex13-04.jsp

성별과 좋아하는 과목 조사

1. 성별을 선택하세요.
☒ 남자 ☐ 여자

2. 좋아하는 과목을 선택하세요.
☒ SQL응용
☐ DB개발도구
☒ JSP
☐ 오라클실무
☐ ERP구축



ex13-04.jsp http://localhost:8080/jspStudy/ch13/ex13-0...

http://localhost:8080/jspStudy/ch13/ex13-04-1.jsp

성별과 좋아하는 과목은

당신은 남자 이고,

좋아하는 과목으로

- SQL응용
- JSP
- ERP구축

을 선택하였습니다.

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>성별과 과목선택 입력화면</title>
8 </head>
9 <body>
10 <h4>성별과 좋아하는 과목 조사</h4>
11 <form method=post action=ex13-04-1.jsp>
12   1.성별을 선택하세요.<br>
13   <input type="radio" name=s1 value="남자" checked>남자&nbsp;  
14   <input type="radio" name=s1 value="여자">여자<p>
15
16   2. 좋아하는 과목을 선택하세요.<br>
17   <input type="checkbox" name=s2 value="SQL응용" checked>SQL응용<br>
18   <input type="checkbox" name=s2 value="DB개발도구">DB개발도구<br>
19   <input type="checkbox" name=s2 value="JSP" checked>JSP<br>
20   <input type="checkbox" name=s2 value="오라클실무">오라클실무<br>
21   <input type="checkbox" name=s2 value="ERP구축">ERP구축<p>
22   <input type=submit value="화 인">
23   <input type=reset value="취 소">
24 </form>
25 </body>
26 </html>

```

ex06-03.jsp 복사

exl3-04-1.jsp

varStatus="vs" 추가.
-를 \${vs.count}로 변경

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <% request.setCharacterEncoding("utf-8"); %>
5
6 <h4> 성별과 좋아하는 과목은 </h4>
7     당신은 <b>${param.s1}</b> 이고,<p>
8     좋아하는 과목으로<br><b>
9     <c:forEach var="ck" items="${paramValues.s2}">
10         - ${ck}<br>
11     </c:forEach>
12     </b><br>을 선택하였습니다.

```

EL식으로

TEL 식으로

2.8 <c:forTokens> 태그

- ▶ <c:forTokens/>
 - ▶ java.util.StringTokenizer를 이용
 - ▶ 구분자(delims)에 맞추어 반복 처리

형식

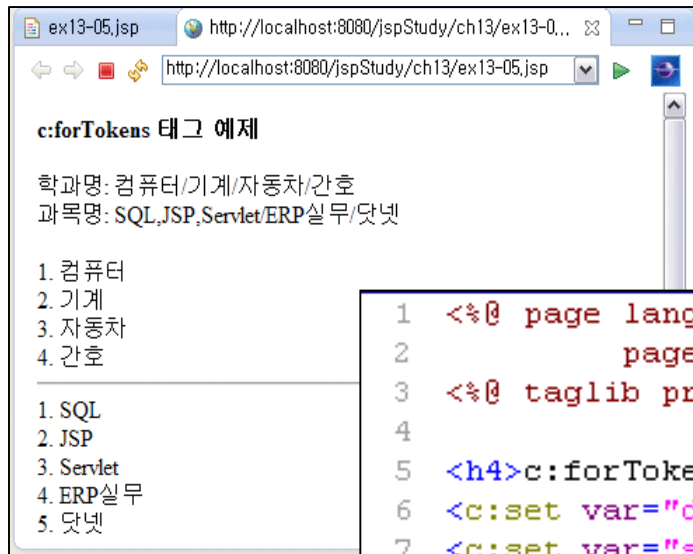
```
<c:forTokens items="문자열값" delims="구분자집합"
              [var="변수명"] [varStatus="상태변수명"]
              [begin="시작값"] [end="최종값"] [step="단계"]>
  body content
</c:forTokens>
```

속성	설 명	속성	설 명
items	배열, 컬렉션(collection), Map 등	begin	시작값
delims	구분자 기호	end	최종값
var	변수명	step	단계값
varStatus	반복상태 값을 갖고 있는 변수		

【예제 13.5】

학과코드의 값("컴퓨터, 기계, 자동차, 간호")과
과목명("SQL,JSP,Servlet/ERP실무/닷넷")을 변수로 선언하고,
구분자(, 또는 /)를 이용하여 출력하는 프로그램을 작성하시오.

생략



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4
5 <h4>c:forTokens 태그 예제</h4>
6 <c:set var="dept" value="컴퓨터/기계/자동차/간호" />
7 <c:set var="subj" value="SQL,JSP,Servlet/ERP실무/닷넷" />
8
9 학과명: ${dept} <br>
10 과목명: ${subj}<p>
11 <c:forTokens var="i" items="${dept}" delims="/" varStatus="vs" >
12     <c:out value="${vs.count}. ${i}"/><br>
13 </c:forTokens>
14 <hr/>
15 <c:forTokens var="name" items="${subj}" delims="," varStatus="vs">
16     ${vs.count}. ${name}<br>
17 </c:forTokens>
```

반복횟수를 count로 출력

JSTL core의 url 관리 태그

▶ url 관리 태그

▶ `<c:import/>`

- ▶ `<jsp:include>` 태그처럼 사용
- ▶ 동적/정적 웹페이지 포함시킴

▶ `<c:url/>`

- ▶ 주소(URL)를 자동으로 생성하여 변수에 저장

▶ `<c:redirect/>`

- ▶ `response.sendRedirect()`를 대체하는 태그
- ▶ 지정 웹페이지로 이동

2.9 <c:import> 태그

- ▶ <c:import/>
 - ▶ 다른 프로그램을 포함시킴
 - ▶ <jsp:include> 액션태그와 동일


태그 1	<pre><c:import url="url" [context="context"] [var="변수명"] [scope="속성"] [charEncoding="인코딩"> Body content <c:param> 서브태그 </c:import></pre>
------	--

태그 2	<pre><c:import url="url" [context="context"] [charEncoding="인코딩"> Body content </c:import></pre>
------	---

속 성	설 명
context	컨텍스트 경로
<c:param>	전송할 파라메타 이름과 값을 지정

2.9 <c:import> 태그

코딩 예 1: "http://localhost:8080/1105001/HelloWorld.jsp"를 url로 불러오고,
이를 data 변수로 지정하여 출력

- `<c:import url="http://localhost:8080/1105001/HelloWorld.jsp" var="data" />`
`${data }`
- 

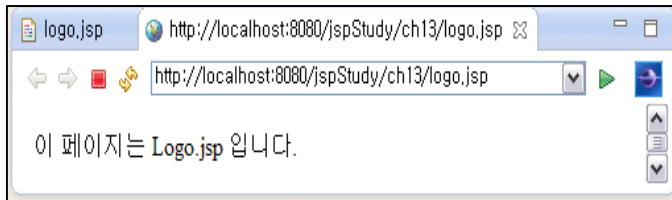
코딩 예 2: "test1.jsp" 프로그램을 포함

- `<c:import url="test1.jsp" />`

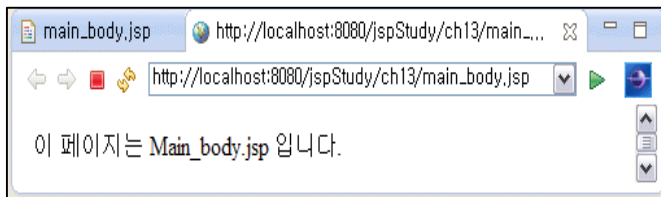
코딩 예 3: 동일 컨텍스트에 있는 자원을 포함

- `<c:import url="/copyright.html" />`
- `<c:import url="http://localhost:8080/0105001/ch10/logo.jsp?id=jskang" />`
- `<c:import url="/logo.jsp" context="/0105002" />`

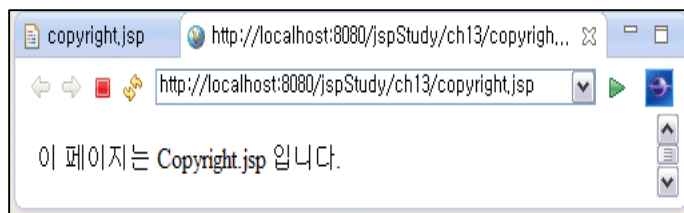
【예제 13.6】예제 13.6의 출력 결과와 같이 "logo.jsp",
"main_body.jsp", "copyright.jsp" 문서를 각각 작성하시오.



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 이 페이지는 Logo.jsp 입니다.
```



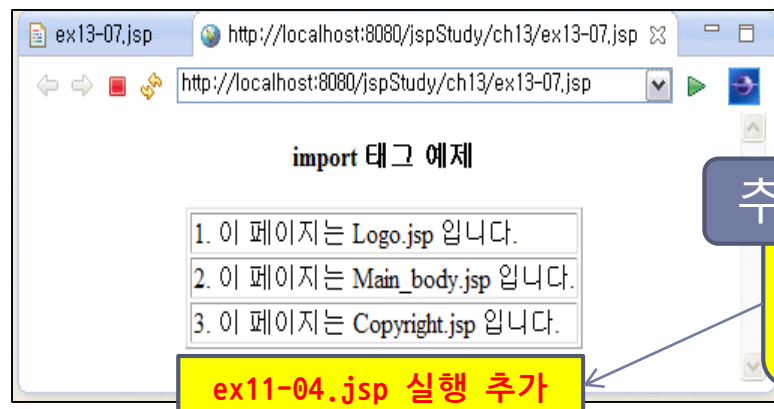
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 이 페이지는 Main_body.jsp 입니다.
```



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 이 페이지는 Copyright.jsp 입니다.
```

【 예제13.7 】

예제 13.6에서 작성한 "logo.jsp", "main_body.jsp", "copyright.jsp" 문서를 이용하여 예제 13.7의 출력하는 프로그램을 작성하시오.



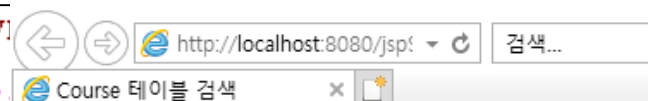
추가문제

ch11 폴더의 ex11-04.jsp를 추가하여 보시오.
(상대경로 : ' ../ch11/ex11-04.jsp ')

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4
5 <center><h4> import 태그 예제 </h4>
6 <table border=1>
7     <tr>
8         <td>1. <c:import url="logo.jsp" %>
9     </tr>
10    <tr>
11        <td>2. <c:import url="main_body.jsp" %>
12    </tr>
13    <tr>
14        <td>3. <c:import url="copyright.jsp" %>
15    </tr>
16 </table></center>

```



import 태그 예제

1. 이 페이지는 Logo.jsp 입니다.
2. 이 페이지는 Main_body.jsp 입니다.
3. 이 페이지는 Copyright.jsp 입니다.

Course 테이블				
순번	과목코드	과목명	학점수	교수번호
1	L1012	웹디자인	2	null
2	L1031	SQL	3	P12
3	L1042	Delphi	3	P13
4	L1052	전자상거래	3	P22
5	L2031	게임이론	3	P23

【 예제13.7 】 추가문제

- ▶ ch11 폴더의 ex11-04.jsp를 추가하여 보시오.
 - ▶ (상대경로 : '../ch11/ex11-04.jsp')

Course 테이블 검색

import 태그 예제

1. 이 페이지는 Logo.jsp 입니다.
2. 이 페이지는 Main_body.jsp 입니다.
3. 이 페이지는 Copyright.jsp 입니다.

Course 테이블					
순번	과목코드	과목명	학점수	교수번호	추가수강료
1	L1012	웹디자인	2	null	20000
2	L1031	SQL	3	P12	30000
3	L1042	Delphi	3	P13	50000
4	L1052	전자상거래	3	P22	30000
5	L2031	게임이론	3	P23	50000

```

ex13-07.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4
5 <center><h4>import 태그 예제</h4>
6   <table border=1>
7     <tr>
8       <td>1. <c:import url="logo.jsp" /></td>
9     </tr>
10    <tr>
11      <td>2. <c:import url="main_body.jsp" /></td>
12    </tr>
13    <tr>
14      <td>3. <c:import url="copyright.jsp" /></td>
15    </tr>
16    <tr>
17      <td><c:import url="../ch11/ex11-04.jsp" /></td>
18    </tr>
19  </table>
20 </center>

```

2.10 <c:redirect> 태그

- ▶ <c:redirect>
 - ▶ 지정한 페이지로 이동
 - ▶ response.sendRedirect(), <jsp:forward>와 동일

태그 1	<code><c:redirect url="url" [context="컨텍스트"]/></code>
------	--

태그 2	<code><c:redirect url="url" [context="컨텍스트"]></code> <code><c:param> 서브타입</code> <code></c:redirect></code>
------	---

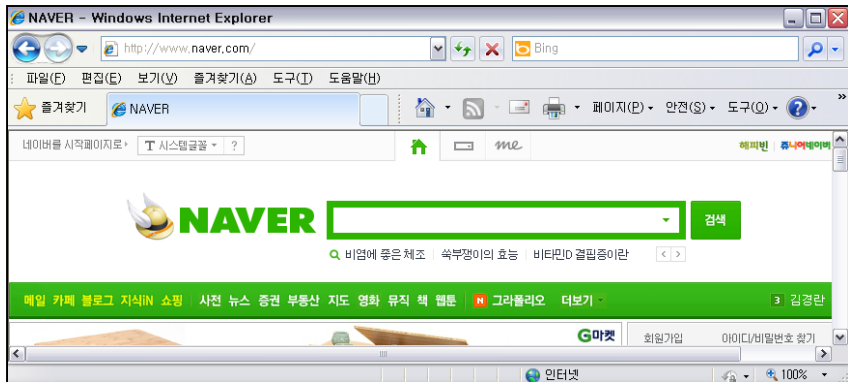
코딩 예 1: test1.jsp 페이지로 이동하고, num1 변수에 5의 값을 전달

- `<c:redirect url="test1.jsp">`
`<c:param name="num1" value="5" />`
`</c:redirect>`

코딩 예 2: "http://www.ync.ac.kr" 지정한 url로 이동

- `<c:redirect url="http://www.ync.ac.kr" />`

【예제 13.8】 naver 검색 사이트("http://www.naver.com") 로 이동하는 프로그램을 작성하시오.



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <h4> c:redirect 태그 예제 </h4>
5 <c:redirect url="http://www.naver.com" />
```

2.11 <c:url> 태그

▶ <c:url>

- ▶ 주소(URL)를 자동으로 생성하여 변수에 저장

태그 1	<code><c:url value="url주소" [var="변수명"] [scope="속성"] /></code>
태그 2	<code><c:url value="url주소" [var="변수명"] [scope="속성"] ></code> <code> <c:param> 서브타입</code> <code></c:url></code>

2.11 <c:param> 태그

- ▶ <c:param>
 - ▶ <c:url>의 서브태그
 - ▶ <jsp:param> 태그와 동일

서브태그 1	<code><c:param name="변수명" value="값" /></code>
--------	---

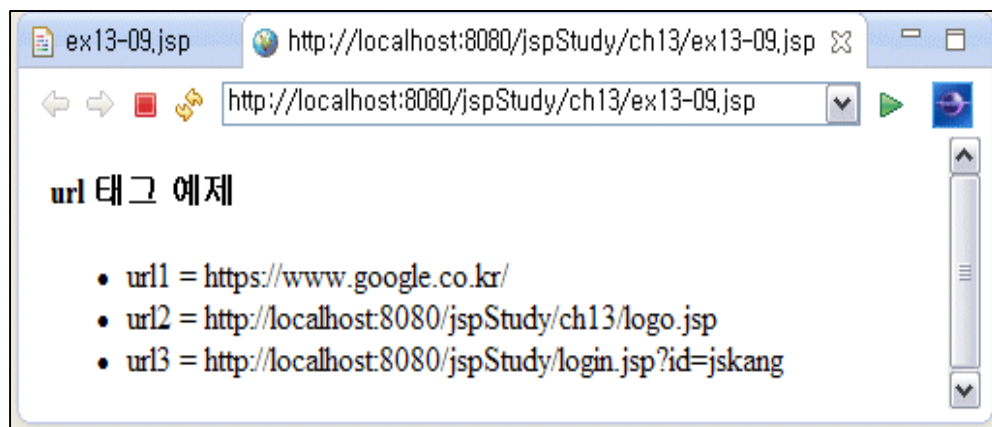
서브태그 2	<code><c:param name="변수명"> 파라메타 값 </c:param></code>
--------	---

코딩 예: test.jsp를 myURL 변수로 지정하고, num 변수에 5를 전달

● `<c:url var="myURL" value="test.jsp">`
 `<c:param name="num" value="5" />`
 `</c:url>`

myURL=test.jsp?num=5

【예제 13.9】 예제 13.9의 출력 결과와 같이 컨텍스트(/), logo.jsp, login.jsp에 id 값으로 "jskang"를 전달하는 url을 생성하여 출력하는 프로그램을 작성하시오.



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4
5 <h4> url 태그 예제 </h4>
6 <c:url var="url1" value="https://www.google.co.kr/" />
7 <c:url var="url2" value="http://localhost:8080/jspStudy/ch13/logo.jsp" />
8 <c:url var="url3" value="http://localhost:8080/jspStudy/login.jsp">
9     <c:param name="id" value="jskang"/>
10 </c:url>
11 <ul>
12     <li>url1 = ${url1}
13     <li>url2 = ${url2}
14     <li>url3 = ${url3}
15 </ul>
```

실행후 구글사이트(url1)로 이동 태그 추가 (<c:redirect> 태그 사용)

3. Format 라이브러리

▶ JSP 프로그램에서 선언

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

▶ 태그라이브러리 지시자 목록

기 능	태 그	설 명
수치 형식	<code><fmt:formatNumber></code>	숫자 데이터를 문자열 양식으로 변환
	<code><fmt:parseNumber></code>	문자열 데이터를 숫자로 변환
시간대 설정	<code><fmt:timeZone></code>	GMT기준으로 지역시간대 설정
	<code><fmt:setTimeZone></code>	시간대 정보를 특정 변수에 저장
날짜와 시간	<code><fmt:formatDate></code>	날짜 데이터를 문자열 양식으로 변환
	<code><fmt:parseDate></code>	문자형 데이터를 Date 객체로 변환
지역 설정	<code><fmt:setLocale></code>	언어를 지정하는 태그
다국어 메시지처리	<code><fmt:Bundle></code>	Properties 파일의 리소스를 불러올 때
	<code><fmt:setbundle></code>	페이지에 사용할 수 있는 번들 지정
	<code><fmt:message></code>	Bundle 태그에서정한 값을 읽을 때
파라메타전달	<code><fmt:param></code>	파라메타 전달
인코딩	<code><fmt:requestEncoding></code>	Request부터 전달받은 값들의 인코딩

13.1 국제화 관련 태그

- ▶ 지역 설정
 - ▶ `<fmt:setLocale>` 태그
- ▶ 시간대 설정
 - ▶ `<fmt:timeZone>` 태그
 - ▶ `<fmt:setTimeZone>` 태그
- ▶ 다국어 처리
 - ▶ `<fmt:bundle>` 태그
 - ▶ `<fmt:setBundle>` 태그
 - ▶ `<fmt:message >` 태그

13.1 국제화 관련 태그

- ▶ `<fmt:setLocale>` 지역설정 태그
 - ▶ 현재 페이지나 특정 영역에 지역정보를 지정
 - ▶ `value`의 "locale"은 "언어코드, 국가 코드"의 조합으로 표기
 - ▶ 한국의 한국어는 "ko_KR"
 - ▶ 미국의 영어는 "en_US"
 - ▶ 일본지역의 일본어는 "ja_JP" 등

태그	<code><fmt:setLocale value="locale" [variant="variant"][scope="속성"] /></code>
----	---

코딩 예: 한국지역으로 한국어를 지정할 때 `value` 속성으로 "ko_KR"을 지정

● `<fmt:setLocale value="ko_KR"/>`

13.1 국제화 관련 태그

▶ 시간대설정 태그

▶ `<fmt:timeZone>` 태그

- ▶ GMT 기준으로 value 속성에 특정 영역에 표준시간대 지정

태그	<code><fmt:timezone value="timeZone"> body content </fmt:timezone></code>
----	---

코딩 예. 한국(Korea Seoul) 시간대로 지정

- `<fmt:timezone value="Korea/Seoul">`
 `<fmt:formatDate value="<%= new Date() %>" />`
 `</fmt:timezone>`

▶ `<fmt:setTimeZone>` 태그

- ▶ 표준 시간대 지정 또는 특정영역에 표준 시간대를 변수에 저장

태그	<code><fmt:setTimeZone value="timeZone" [var="변수명"][scope="속성"] /></code>
----	---

코딩 예. 한국(Korea Seoul)을 세션 영역에 표준 시간대로 지정

- `<fmt:setTimeZone value="Korea/Seoul" var="localZone" scope="session" />`

3.2 다국어 지원 태그

- ▶ 다국어 지원 태그 3가지
 - ▶ `<fmt:bundle>` 태그 : 번들 지정 태그
 - ▶ `<fmt:setBundle>` 태그 : 번들 지정 또는 값을 저장
 - ▶ `<fmt:message>` 태그의 세 가지 : 번들의 키 값을 읽어 출력
- ▶ 값 전달 : 키에 값을 전달
 - ▶ `<fmt:param>` 서브 태그
- ▶ 번들(bundle)이란?
 - ▶ "묶음" 의미
 - ▶ "key"와 "key 값"이 여러 개 저장된 정보
 - ▶ bundle 파일명
 - ▶ 번들명과 언어, 국가코드, properties순으로 구성
 - ▶ 국가코드는 생략 가능

3.2 다국어 지원 태그

▶ 번들에 사용할 프로퍼티 파일 생성하기

생략

- ▶ 번들에 사용할 프로퍼티의 텍스트 파일은 native2ascii.exe를 실행하여 아스키(ascii) 코드로 변환하여 프로퍼티(properties) 파일을 생성
- ▶ native2ascii.exe 파일은 jdk가 설치된 bin 폴더

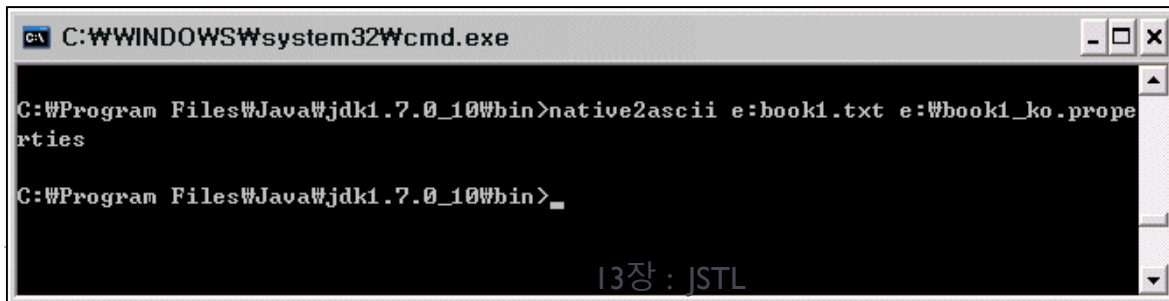
태그

native2ascii.exe 소스텍스트파일.txt 프로퍼티파일명.properties

▶ properties 파일을 생성하는 순서

생략

- ▶ ① 편집기를 이용하여 text.txt, text11.txt 텍스트 번들 파일 생성
- ▶ ② [시작][실행] 메뉴를 클릭하여 "cmd"를 입력한 후 [확인] 버튼을 클릭
- ▶ ③ jdk가 설치된 bin 하위폴더로 이동. (cd 폴더명)
- ▶ ④ native2ascii.exe *.txt *.properties 형식으로 변환
- ▶ ⑤ 확장자가 .properties인 파일을 프로젝트의 src 폴더에 복사



```
C:\WINDOWS\system32\cmd.exe

C:\Program Files\Java\jdk1.7.0_10\bin>native2ascii e:\book1.txt e:\book1_ko.properties

C:\Program Files\Java\jdk1.7.0_10\bin>
```


영문(text.txt)과 한글(text1.txt) 번들 파일을 각각 생성하여
【 예제 13.10 native2ascii.exe 파일로 변환하고, 변환한 파일을 src 폴더에 복사하시오. 단, 키명을 national, language, job순임. **생략**

```
1 national=America
2 language=English
3 job=Web Programmer
4
```

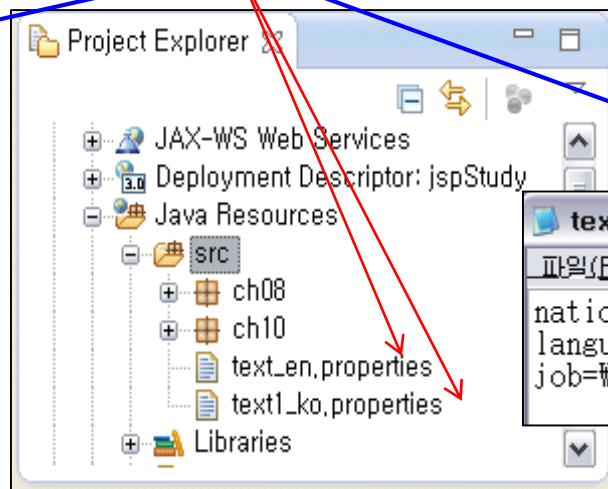
```
1 national=대한민국
2 language=한국어
3 job=웹프로그래머
```

native2ascii.exe *.txt *.properties 형식으로 변환

text_en.properties - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

national=America
language=English
job=Web Programmer



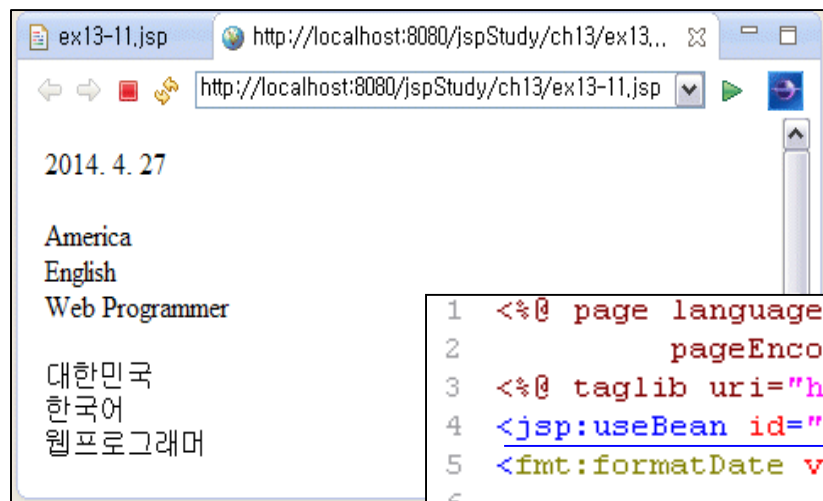
text1_ko.properties - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

national=똥ub300똥ud55c똥ubbfcd똥uad6d
language=똥ud55c똥uad6d똥uc5b4
job=똥uc6f9똥ud504똥ub85c똥uadf8똥ub798똥uba38

[예제 13.11] 예제 13.10에서 생성한 번들 프로퍼티 파일을 이용하여 출력하는 프로그램을 작성하시오. [결과가 나오지 않을 경우, [Project][clean] 메뉴를 클릭하고, "새로고침" 버튼을 클릭한다.

생략



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
4 <jsp:useBean id="now" class="java.util.Date" />
5 <fmt:formatDate value="{now}" /><p>
6
7 <fmt:setLocale value="en_US"/>
8 <fmt:bundle basename="text">
9     <fmt:message key="national" /><br>
10    <fmt:message key="language" /><br>
11    <fmt:message key="job" /><p>
12 </fmt:bundle>
13
14 <fmt:setLocale value="ko_KR"/>
15 <fmt:bundle basename="text1">
16     <fmt:message key="national" /><br>
17     <fmt:message key="language" /><br>
18     <fmt:message key="job" />
19 </fmt:bundle>
20
```

3.3 수치 형식의 태그(1)

1) 숫자 형식의 <fmt:formatNumber> 태그

- ▶ 숫자 데이터를 양식에 맞춰 문자열로 변환

태그 1

```
<fmt:formatNumber value="숫자형 값"
                  [type="{number | currency | percent}"]
                  [pattern="사용자패턴"]
                  [currencyCode="통화코드"]
                  [currencySymbol="통화기호"]
                  [groupingUsed="{true|false}"]
                  [maxIntegerDigits="정수형최대자리수"]
                  [minIntegerDigits="정수형최소자리수"]
                  [maxFractionDigits="소수최대자리수"]
                  [minFractionDigits="소수최소자리수"]
                  [var="변수명"] [scope="속성"] />
```

기호	설 명	기호	설 명
0	한자리 대체 숫자	-	음수의 접두어 표시
E	지수 형태 표현	%	100을 곱한 백분율로 표시
#	한자리 대체 숫자	?	100을 곱한 밀리(mille)로 표시
.	소숫점 구분 표시	¤	통화기호 표시
,	그룹핑 구분자	X	다른 문자로 접두어나 접미어 표현
;	형식 구분	'	접두어나 접미어 안에 ' 특수문자 사용

태그 2

```
<fmt:formatNumber [type="{number | currency | percent}"
    [pattern="사용자패턴"]
    [currencyCode="통화코드"]
    [currencySymbol="통화기호"]
    [groupingUsed="{true|false}"]
    [maxIntegerDigits="정수형최대자리수"]
    [minIntegerDigits="정수형최소자리수"]
    [maxFractionDigits="소수최대자리수"]
    [minFractionDigits="소수최소자리수"]
    [var="변수명"] [scope="속성"]
    "형식화할 숫자형 값"
</fmt:formatNumber>
```

속 성	설 명
value	양식에 출력한 숫자형 데이터
type	number(기본값), currency, percent
pattern	출력할 양식 패턴 지정 (표13.15) 참조
currencyCode	ISO 4217 통화 코드 지정. 통화 형식일 때만 적용
currencySymbol	통화 기호 지정. 통화 형식일 때만 적용
groupingUsed	출력에 그룹 분리 기호를 포함할지 여부를 지정
maxIntegerDigits	출력에서 integer 최대 자릿수 지정
minIntegerDigits	출력에서 integer 최소 자릿수 지정
maxFractionDigits	출력에서 소수점 이하 최대 자릿수 지정
minFractionDigits	출력에서 소수점 이하 최소 자릿수 지정
var	출력 결과 문자열을 담는 scope에 해당하는 변수명
scope	var의 scope 지정, 기본값은 page

3.3 수치 형식의 태그[2]

- ▶ 숫자형식의 <fmt:parseNumber> :태그
 - ▶ 문자형 데이터를 숫자형 데이터로 변환

태그 1	<pre><fmt:parseNumber value="문자형 데이터" [type="{number currency percent}"] [pattern="사용자패턴"] [parseLocale="기본형식패턴"] [IntegerOnly="{true false}"] [groupingUsed="{true false}"] [var="변수명"] [scope="속성"] /></pre>
---------	--

태그 2	<pre><fmt:parseNumber [type="{number currency percent}"] [pattern="사용자패턴"] [parseLocale="기본형식패턴"] [IntegerOnly="{true false}"] [groupingUsed="{true false}"] [var="변수명"] [scope="속성"] 변환할 문자형 데이터 </fmt:parseNumber></pre>
---------	---

[예제 13.12]

예제 13.12의 출력결과와 같이 숫자형 데이터를 여러 형태로 변환하여 출력하는 프로그램을 작성하시오.

ex13-12.jsp JSTL fmt:formatNumber Tag

http://localhost:8080/jspStudy/ch13/ex13-12.jsp

숫자형식 변환 예제			
숫자 데이터 ==>		12345678.1234	
1. 화폐 타입	₩ 12,345,678.12	2. USA 화폐	\$12,345,678.12
3. 정수 3자리	678.123	4. 소수점 3자리	12,345,678.123
5. 3자리 퍼센트	812%	6. 10자리 퍼센트	1,234,567,812.3400002000%
7. 그룹분리기호없음	12345678.123	8. 지수 타입	1.235E7

9. 한국 화폐 출력 추가

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5
6 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "">
7 <html>
8 <head>
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10 <title>JSTL fmt:formatNumber Tag</title>
11 </head>
12 <body>
13 <c:set var="num1" value="12345678.1234"/>
14 <table border=1>
```

```
15 <tr>
16   <th colspan=4>숫자형식 변환 예제</th>
17 </tr>
18 <tr align=center>
19   <td colspan=2>숫자 데이터 ==></td>
20   <td colspan=2>${num1}</td>
21 </tr>
22 <tr>
23   <td>1. 화폐 타입</td>
24   <td><fmt:formatNumber value="${num1}" type="currency"/></td>
25   <td>2. USA 화폐</td>
26   <td><fmt:setLocale value="en_US"/>
27     <fmt:formatNumber value="${num1}" type="currency"/></td>
28 </tr>
29 <tr>
30   <td>3. 정수 3자리</td>
31   <td><fmt:formatNumber type="number"
32     maxIntegerDigits="3" value="${num1}"/></td>
33   <td>4. 소수점 3자리</td>
34   <td><fmt:formatNumber type="number"
35     maxFractionDigits="3" value="${num1}"/></td>
36 </tr>
37 <tr>
38   <td>5. 3자리 퍼센트</td>
39   <td><fmt:formatNumber type="percent"
40     maxIntegerDigits="3" value="${num1}"/></td>
41   <td>6. 10자리 퍼센트</td>
42   <td><fmt:formatNumber type="percent"
43     minFractionDigits="10" value="${num1}"/></td>
44 </tr>
45 <tr>
46   <td>7. 그룹분리 타입</td>
47   <td><fmt:formatNumber type="number"
48     groupingUsed="false" value="${num1}"/></td>
49   <td>8. 지수 타입 </td>
50   <td><fmt:formatNumber type="number"
51     pattern="#.###E0" value="${num1}"/></td>
52 </tr>
53 </table>
54 </body>
55 </html>
```

3.4 날짜 형식의 태그

1번 할
차례

- ▶ <fmt:formatDate> : 날짜 데이터를 문자열 형식으로 변환

태그	<pre><fmt:formatDate value="날짜형" [type="{time date both}" [dateStyle="{default sort medium long full}" [timeStyle="{default sort medium long full}" [pattern = "양식패턴"] [timeZone="timeZone"] [var="변수명"] [scope="속성"] /></pre>
----	--

속 성	텍스트를 입력하십시오	설 명
value	양식에 출력한 날짜형 데이터	
type	date , time, both	
pattern	사용자 형식의 스타일 지정	
dateStyle	type 생략, date나 both 일 때 미리 정의된 날짜 형식 지정	
timeStyle	type이 time 또는 both 일 때, 미리 정의된 시간 형식 지정	
timeZone	java.util.TimeZone 형식으로 된 시간에 나타날 타임 존 지정	
var	출력 결과 문자열을 담는 scope에 해당하는 변수명 지정	
scope	var의 scope 지정	

3.4 날짜 형식의 태그

- ▶ `<fmt:parseDate>` : 문자형 데이터를 Date 객체로 변환

태그 1	<pre> <fmt:parseDate value="날짜형식 문자열" [type="{time date both}"] [dateStyle="{default sort medium long full}"] [timeStyle="{default sort medium long full}"] [pattern ="양식패턴"] [timeZone="timeZone"] [parseLocale="parseLocale"] [var="변수명"] [scope="속성"] /> </pre>
태그 2	<pre> <fmt:parseDate [type="{time date both}"] [dateStyle="{default sort medium long full}"] [timeStyle="{default sort medium long full}"] [pattern ="양식패턴"] [timeZone="timeZone"] [parseLocale="parseLocale"] [var="변수명"] [scope="속성"] 변환할 날짜형 값 </fmt:parseDate> </pre>

【예제 13.13】 예제 13.13의 출력결과와 같이 다양한 날짜 데이터를 출력하는 프로그램을 작성하시오.

ex13-13.jsp JSTL fmt:dateNumber Tag

http://localhost:8080/jspStudy/ch13/ex13-13.jsp

날짜형 형식 변환 예제			
날짜 데이터 ==>		Tue Nov 24 11:45:02 KST 2020	
1.시간 타입	오전 11:45:02	2.날짜 타입	2020. 11. 24
3.날짜와 시간 타입	2020. 11. 24 오전 11:45:02	4.날짜와 시간 (short)	20. 11. 24 오전 11:45
5.날짜와 시간 (medium)	20. 11. 24 오전 11:45:02	6.날짜와 시간 (long)	20. 11. 24 오전 11시 45분 02초
7.날짜와 시간 (full)	20. 11. 24 오전 11시 45분 02초 KST	8.사용자 지정 형식	2020-11-24

```
<jsp:useBean id="now" class="java.util.Date" />
```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5
6 <!-- JSTL 4.01 Transitional//EN "" -->
7
8 <head>
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10 <title>JSTL fmt:dateNumber Tag</title>
11 </head>
12 <body>
13 <c:set var="now" value="<%=new java.util.Date()%>" />
14 <table border=1>
15 <tr>
16 <th colspan=4>날짜형 형식 변환 예제</th>
17 </tr>

```

```

18 <tr align=center>
19     <td colspan=2>날짜 데이터 ==></td>
20     <td colspan=2>${now}</td>
21 </tr>
22 <tr>
23     <td>1.시간 타입</td>
24     <td><fmt:formatDate type="time" value="${now}" /></td>
25     <td>2.날짜 타입</td>
26     <td><fmt:formatDate type="date" value="${now}" /></td>
27 </tr>
28 <tr>
29     <td>3.날짜와 시간 타입</td>
30     <td><fmt:formatDate type="both" value="${now}" /></td>
31     <td>4.날짜와 시간 (short)</td>
32     <td><fmt:formatDate type="both"
33         dateStyle="short" timeStyle="short" value="${now}" /></td>
34 </tr>
35 <tr>
36     <td>5.날짜와 시간 (medium)</td>
37     <td><fmt:formatDate type="both"
38         dateStyle="short" timeStyle="medium" value="${now}" /></td>
39     <td>6.날짜와 시간 (long)</td>
40     <td><fmt:formatDate type="both"
41         dateStyle="short" timeStyle="long" value="${now}" /></td>
42 </tr>
43 <tr>
44     <td>7.날짜와 시간 (full)</td>
45     <td><fmt:formatDate type="both"
46         dateStyle="short" timeStyle="full" value="${now}" /></td>
47     <td>8.사용자 지정 형식</td>
48     <td><fmt:formatDate pattern="yyyy-MM-dd" value="${now}" /></td>
49 </tr>
50 </table>
51 </body>
52 </html>

```

3.5 <fmt:requestEncoding> 인코딩 태그

- ▶ <fmt:requestEncoding> 인코딩 태그
 - ▶ request의 문자 인코딩
 - ▶ <% request.setCharacterEncoding("utf-8"); %>와 동일

태그	<fmt:requestEncoding [value="문자인코딩"] />
----	---

코딩 예. <% request.setCharacterEncoding("utf-8"); %>의 표현식으로 코딩한 스크립팅 태그를 <fmt:requestEncoding> 인코딩 태그로 변경하시오.

● <fmt:requestEncoding value="utf-8" />

	1	2	3	4	5	6
1	<%@ page language="java" contentType="text/html; charset=UTF-8"					
2	pageEncoding="UTF-8"%>					
3	<% request.setCharacterEncoding("UTF-8"); %>					

4. SQL 라이브러리

- ▶ JSP 프로그램에서 선언
 - ▶ `<%@ taglib prefix=sql uri=http://java.sun.com/jsp/jstl/sql"%>`
- ▶ SQL라이브러리 태그

태 그	설 명
<code><sql:setDataSource></code>	데이터베이스 서버 및 계정 등 지정
<code><sql:query></code>	SELECT문 수행
<code><sql:update></code>	INSERT/UPDATE/DELETE문 수행
<code><sql:param></code>	SQL문의 ? 값의 기술
<code><sql:transaction></code>	트랜잭션 설정
<code><sql:dateParam></code>	<code><sql:query></code> 와 <code><sql:update></code> 태그의 서브 태그



JDBC 프로그래밍 절차 사용 안함.

4.1 <sql:setDataSource> 태그

- ▶ <sql:setDataSource> 태그 ← context.xml 대체 태그
 - ▶ 데이터베이스 서버의 url, jdbc 드라이버, 계정, 암호 등의 속성
 - ▶ JDBC 프로그래밍 절차의 2단계와 3단계 대체 태그

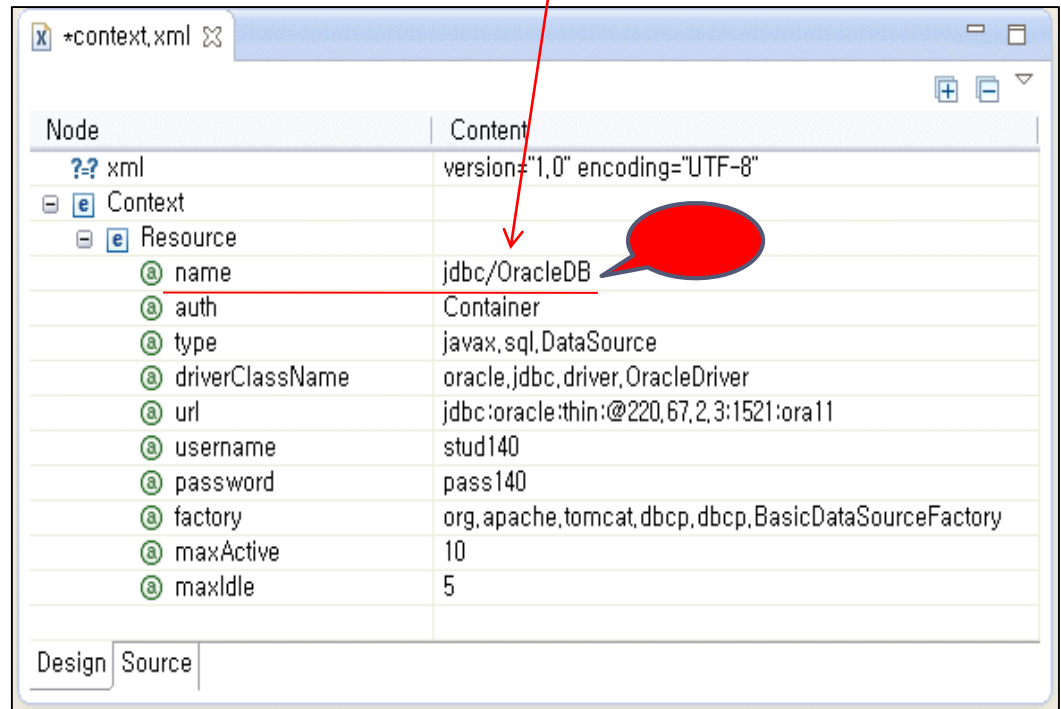
태그	<code><sql:setDataSource [dataSource="dataSource" url="jdbcUri"] [driver="driverClassName"] [user="사용자명"] [password="암호"] [var="변수명"] [scope="속성] /></code>
----	--

코딩 예:

```
● <sql:setDataSource    url="jdbc:oracle:thin:@222.66.2.3:1521:ora11"  
                        driver="oracle.jdbc.driver.OracleDriver"  
                        user="stud"  
                        password="pass"  
                        var="ora11"  
                        scope="page"    />
```

4.1 <sql:setDataSource> 태그

- ▶ <sql:setDataSource> 태그 **생략 방법**
 - ▶ 예제 11.1 context.xml 파일 사용
 - ▶ 리소스명 : jdbc/OracleDB
 - ▶ <sql:query>와 <sql:update> 태그의 dataSource 속성값



4.2 <sql:query> 태그

- ▶ <sql:query> 태그
 - ▶ SELECT문 실행 태그
 - ▶ JDBC 프로그래밍 절차의 4단계와 5단계

2, 3단계 생략함

context.xml (DBCP)

태그 1	<pre><sql:query sql="sqlQuery" var="변수명" [scope="속성"] [dataSource="dataSource"] [maxRow="최대행"] [startRow="시작행"] /></pre>
태그 2	<pre><sql:query sql="sqlQuery" var="변수명" [scope="속성"] [dataSource="dataSource"] [maxRow="최대행"] [startRow="시작행"]> <sql:param>actions </sql:query></pre>
태그 3	<pre><sql:query var="변수명" [scope="속성"] [dataSource="dataSource"] [maxRow="최대행"] [startRow="시작행"]> SELECT Statement optional <sql:param> actions </sql:query></pre>

? 값 지정

4.2 <sql:query> 태그

속성	동적값	타입	설명
sql	true	String	SQL 질의어
dataSource	True	String	데이터소스
maxRow	True	Int	질의어 결과로 가져올 최대 행의 수
startRow	True	Int	질의어 결과로 가져올 시작 행
Var	True	String	질의어 결과를 가져올 변수
scope	true	String	변수의 scope

<sql:query> 태그 실행후 변수참조 프로퍼티

프로퍼티	설 명
rows	SortedMap 객체의 배열. 칼럼명으로 ResultSet 행과 매핑
rowsByIndex	각각 ResultSet의 행과 상응.
columnNames	ResultSet에서 칼럼명의 스트링 배열.
rowCount	쿼리 결과에 있는 전체 행의 수
limitedByMaxRows	maxRows 속성의 값에 의해 쿼리가 제한될 경우 true.

태그	<code><sql:query var="변수명" [dataSource="dataSource"]</code> SELECT Statement <code></sql:query></code>
----	--

`${변수명.프로퍼티}`

- ✓ `${변수명.rowcount}` : 검색된 전체 행의 수 반환
- ✓ `${변수명.rows}` : 검색된 행 반환
- ✓ `${변수명.columnNames}` : 테이블의 칼럼명 반환

인출 행은 0부터

4.2 <sql:query> 태그

코딩 예. "SELECT * FROM Department" 문을 <sql:query> 태그로 코딩하시오.

● <sql:query var="**rs**" dataSource="**jdbc/OracleDB**">

select * from department

context.xml (DBCP)

</sql:query>

6단계:검색한 값 인출

<c:forEach var="**변수명**" items="**\${rs.columnNames}**">

\${변수명} // 컬럼명 출력

</c:forEach>

columnNames 속성 : 테이블의 컬럼명 인출

<c:forEach var="**row**" items="**\${rs.rows}**">

\${row.dept_id} // **\${변수명.컬럼명}** : 행의 컬럼값

\${row.dept_name}

\$ row.dept_tel}

rows 속성 : 결과 행의 데이터 인출

</c:forEach>

4.2 <sql:update> 태그

2, 3단계 생략함

context.xml (DBCP)

- ▶ <sql:update> 태그
 - ▶ INSERT, UPDATE, DELETE문 실행 태그
 - ▶ JDBC 프로그래밍 절차의 4단계와 5단계

태그 1	<pre><sql:update sql="sqlUpdate" [dataSource="dataSource"] [var="변수명"] [scope="속성"] /></pre>
태그 2	<pre><sql:update sql="sqlUpdate" [dataSource="dataSource"] [var="변수명"] [scope="속성"] <sql:param> actions </sql:update></pre>
태그 3	<pre><sql:update [dataSource="dataSource"] [var="변수명"] [scope="속성"] Insert/Update/Delete statement <sql:param> actions </sql:update></pre>

? 값 지정

속성	동적값	타입	설명
sql	true	String	Insert, Update, Delete문
dataSource	True	String	데이터소스
var	True	String	결과를 저장할 변수
scope	true	String	변수의 scope

코딩 예1: Department 테이블에 한 행을 추가하는 INSERT문

```
● <sql:update dataSource="jdbc/OracleDB">  
    Insert into department (dept_id, dept_name, dept_tel)  
    values ( ' 컴공 ' , ' 컴퓨터공학과 ' , ' 053-450-1234 ' )  
</sql:update>
```

코딩 예2: Department 테이블에서 학과코드가 ' 컴공 ' 인 행을 삭제하는 DELETE문

```
● <sql:update dataSource="jdbc/OracleDB" var="result">  
    delete from department where dept_id = ' 컴공 '  
</sql:update>
```

코딩 예3: Department 테이블에 칼럼값을 수정하는 UPDATE문

```
● <sql:update dataSource="jdbc/OracleDB">  
    update department  
    set dept_name=?  
    where dept_id = ?  
    <sql:param value="${deptname}"/>  
    <sql:param value="${deptid}"/>  
</sql:update>
```

4.4 <sql:param> 태그

- ▶ <sql:param> 태그
 - ▶ SQL 문의 위치지정자(?)에 대한 파라메타 값 지정
 - ▶ <sql:query>와 <sql:update>의 서브 태그
 - ▶ value 속성에 파라메타 값 기술

서브태그 1	<sql:param value="value" />
--------	-----------------------------

서브태그 2	<sql:param> parameter value </sql:param>
--------	---

속성	동적값	타입	설명
value	true	Object	파라메터 값

4.5 <sql:transaction> 태그

▶ <sql:transaction> 태그

- ▶ INSERT문, UPDATE문, DELETE문이 실행될 때 트랜잭션 제어

```
<sql:transaction [dataSource="dataSource"]  
                  [isolation=isolationlevel"]  
    <sql:query> <sql:update>문  
</sql:transaction>
```

“read_committed”
“read_uncommitted”
“repeatable_read”
“serializable”

속성	동적값	타입	설명
isolation	true	String	고립화 레벨
dataSource	True	String	데이터소스

4.6 <sql:dateParam> 태그

- ▶ <sql:dateParam> 태그
 - ▶ <sql:query>와 <sql:update> 태그의 서브 태그
 - ▶ "java.util.Date" 타입으로 SQL문의 파라메타 값 지정

```
<sql:dateParam value="값" type="[date | time | timestamp]" />
```

속성	동적값	타입	설명
value	true	Java.Util.Date	Date, Time, Timestamp 타입
type	true	String	Date, Time, Timestamp 타입 중 하나

SQL 라이브러리 태그의 코딩 순서

SELECT문

`<sql:query var="rs" >` 태그로
SELECT문 실행

`<c:forEach items="${rs.rows}">`
검색된 행 인출 및 처리

INSERT, UPDATE, DELETE문

`<sql:update var="r" >` 태그로
INSERT, UPDATE, DELETE문 실행

변수 r 행의 수 반환 또는 무시

▶ 예제: 데이터소스로 부터 department 테이블을 검색하여 출력하시오.

```
<%@ page language="java" contentType="text/html; charset="UTF-8"
    pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<sql:setDataSource url="jdbc:oracle:thin:@220.67.2.3:1521:ora11"
    driver="oracle.jdbc.driver.OracleDriver"
    user="stud"
    password="pass"
    var="ds1"
    scope="page" />
```

context.xml
대체 태그

```
<sql:query sql="select * from department" var="rs1" dataSource="${ds1}" />
```

```
<html>
  <head><title> Department 테이블 검색</title> </head>
```

```
  <body>
    <table border="1" >
```

```
      <tr>
        <c:forEach var=" cName" items=" ${rs1.columnNames }">
          <td>${cName}</td>
        </c:forEach>
      </tr>
```

```
      <c:forEach var="col" items=" ${rs1.rows}" >
        <tr>
          <td> ${col.dept_id}</td>
          <td> ${col.dept_name}</td>
          <td> ${col.dept_tel}</td>
        </tr>
      </c:forEach>
```

```
</table></body></html>
```

Department 테이블 조회

DEPT ID	DEPT NAME	DEPT TEL
컴공	컴퓨터계열	765-4100
정통	정보통신공학과	765-4200
경영	경영학과	765-4400
행정	세무행정학과	765-4500
대학	대학본부	

[예제 13.14]

Course 테이블에서 3학점인 과목을 검색하여 출력하는 프로그램을 작성하시오. 오라클 DB 서버의 ip 주소는 220.67.2.3, port 번호는 1521, SID는 ora11, 계정은 stud, 암호는 pass임.

context.xml
파일 사용 안함



ex13-14.jsp http://localhost:8080/jspStudy/ch13/ex13-...

http://localhost:8080/jspStudy/ch13/ex13-14.jsp

Course 테이블 조회

번호	과목코드	과목명	학점	추가수강료
1	L1031	SQL	3	30000
2	L1032	자바프로그래밍	3	0
3	L1042	Delphi	3	50000
4	L1052	전자상거래	3	30000
5	L2031	게임이론	3	50000
6	K1112	SQL	3	30000

context.xml
대체 태그

dataSource 속성값

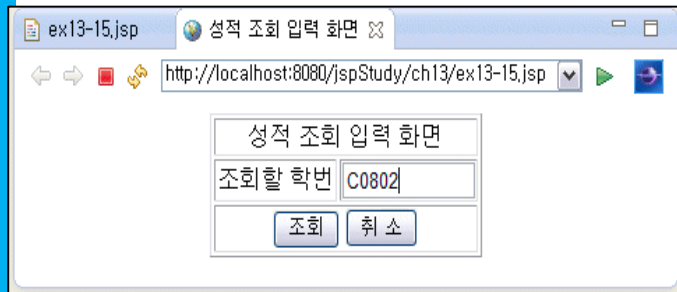
count 속성 : varStatus 변수의 반복 값

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
5
6 <sql:setDataSource
7   url="jdbc:oracle:thin:@220.67.2.3:1521:ora11"
8   driver="oracle.jdbc.driver.OracleDriver"
9   user="stud140"
10  password="pass140"
11  var="ds1"
12  scope="page" />
13
14 <sql:query var="rs" dataSource="${ds1}" >
15   select course_id, title, c_number, nvl(course_fees,0) "fee"
16   from   course
17   where  c_number=3
18 </sql:query>
19
20 <center><h4> Course 테이블 조회 </h4>
21 <table border="1">
22   <tr border="1" align="center">
23     <th>번호</th>
24     <th>과목코드</th>
25     <th>과목명</th>
26     <th>학점</th>
27     <th>추가수강료</th>
28   </tr>
29   <c:forEach var="row" items="${rs.rows}" varStatus="status">
30     <tr>
31       <td align="center">${status.count}</td>
32       <td align="center">${row.course_id}</td>
33       <td>${row.title}</td>
34       <td align="center">${row.c_number}</td>
35       <td align="right">${row.fee}</td>
36     </tr>
37   </c:forEach>
38 </table></center>
```

[예제 13.15]

SG_Scores 테이블을 이용하여 개인별 성적표를 출력하는 프로그램을 작성하시오.

context.xml
파일 사용



개 인 별 성 적 표

학과	학년	학번	성명
컴퓨터계열	3	C0802	서희경

순번	과목번호	과목명	학점	등급
1	L0011	TOEIC연구	2	A+
2	L1031	SQL	3	A+
3	L1032	자바프로그래밍	3	A
4	L1041	컴퓨터네트워크	2	A+
5	L1042	Delphi	3	A+
6	L1051	웹서버관리	2	C+
7	L1052	전자상거래	3	B+

총취득과목수	[7] 과목
총취득학점수	[18] 학점
전체평균평점	[4.03] 점

▶ 등가조인의 SELECT문-1

- ▶ Student 테이블과 Department 테이블로부터 학과명(dept_name), 학년(year), 학번(student_id), 성명(name) 검색

```
SELECT dept_name, year, student_id, name
FROM Student join Department using (Dept_id)
WHERE student_id = '조회할 학번';
```

▶ 등가조인 SELECT문-2

- ▶ SG_Scores 테이블과 Course 테이블로부터 과목코드(course_id), 과목명(title), 학점수(c_number), 등급(grade) 검색

```
select Course_id, title, c_number, grade
from Sg_scores join Course using (Course_id)
where student_id = '조회할 학번';
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>성적 조회 입력 화면</title>
8 <script language="JavaScript">
9     function In_Check() {
10         if(document.hbform.hb.value == "") {
11             alert('학번을 입력하세요!!!');
12             return;
13         }
14         document.hbform.submit();
15     }
16 </script>
17 </head>
18 <body>
19 <center>
20 <form method="post" action="ex13-15-1.jsp" name="hbform">
21 <table border="1">
22 <tr>
23 <td colspan="2" align="center">성적 조회 입력 화면</td>
24 </tr>
25 <tr>
26 <td align="center">조회할 학번 </td>
27 <td><input type="text" name="hb" size="10"></td>
28 </tr>
29 <tr align="center">
30 <td colspan="2">
31 <input type="button" name="modify" value="조회" OnClick= "In_Check()">
32 <input type="reset" value="취 소"></td>
33 </tr>
34 </table>
35 </form></center>
36 </body>
37 </html>
```

ex13-15-1.jsp

context.xml 사용시
<sql:setDataSource> 생략

context.xml (DBCP)

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
5 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
6
7 <c:set var="srow" value="0" />
8 <c:set var="mrow" value="1" />
9 <sql:query var="rs" dataSource="jdbc/OracleDB" maxRows="${mrow}" startRow="${srow}">
10     select dept_name, year, student_id, name
11     from Student join Department using (Dept_id)
12     where student_id =?
13     <sql:param value="${param.hb}" />
14 </sql:query>
15
16 <sql:query var="rs1" dataSource="jdbc/OracleDB">
17     select Course_id, title, c_number, grade
18     from SG_Scores join Course using (Course_id)
19     where student_id =?
20     <sql:param value="${param.hb}" />
21 </sql:query>
22
```

Node	Content
xml	version="1.0" encoding="UTF-8"
Context	
Resource	
name	jdbc/OracleDB
auth	Container
type	javax.sql.DataSource
driverClassName	oracle.jdbc.driver.OracleDriver
url	jdbc:oracle:thin:@220.67.2.3:1521:ora11
username	stud140
password	pass140
factory	org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory
maxActive	10
maxIdle	5


```

24 <html>
25 <head>
26 <title>성적 조회 화면</title>
27 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
28 </head>
29 <body>
30 <table border="1" align="center">
31   <tr border="1" align="center">
32     <td><b>개 인 별 성 적 표 </b></td>
33   </tr>
34 </table><br>
35 <table border="1" align="center">
36   <tr border="1" align="center">
37     <th>학과</th>
38     <th>학년</th>
39     <th>학번</th>
40     <th>성명</th>
41   </tr>
42   <c:forEach items="${rs.rows}" var="rs" >
43     <tr border="1" align="center">
44       <td>${rs.dept_Name}</td>
45       <td>${rs.year}</td>
46       <td>${rs.student_Id}</td>
47       <td>${rs.name}</td>
48     </tr>
49   </c:forEach>
50 </table><br>
51
52 <table border="1" align="center">
53   <tr>
54     <th>순번</th>
55     <th>과목번호</th>
56     <th>과목명</th>
57     <th>학점</th>
58     <th>등급</th>
59   </tr>
60
61   <c:set var="sno" value="${rs1.rowCount}" />
62   <c:set var="sum_Avg" value="0.00" />

```

```

63 <c:forEach items="${rs1.rows}" var="rs1" varStatus="status">
64 <tr border=1>
65 <td align=center>${status.count}</td>
66 <td align=center>${rs1.course_Id}</td>
67 <td>${rs1.title}</td>
68 <td align=center>${rs1.c_Number}</td>
69 <td>${rs1.grade}</td>
70 </tr>
71 <c:choose>
72 <c:when test='${rs1.grade == "A+"}'><c:set var="avg" value="4.5"/></c:when>
73 <c:when test='${rs1.grade == "A"}'><c:set var="avg" value="4.0"/></c:when>
74 <c:when test='${rs1.grade == "B+"}'><c:set var="avg" value="3.5"/></c:when>
75 <c:when test='${rs1.grade == "B"}'><c:set var="avg" value="3.0"/></c:when>
76 <c:when test='${rs1.grade == "C+"}'><c:set var="avg" value="2.5"/></c:when>
77 <c:when test='${rs1.grade == "C"}'><c:set var="avg" value="2.0"/></c:when>
78 <c:when test='${rs1.grade == "D+"}'><c:set var="avg" value="1.5"/></c:when>
79 <c:when test='${rs1.grade == "D"}'><c:set var="avg" value="1.0"/></c:when>
80 <c:when test='${rs1.grade == "F"}'><c:set var="avg" value="0.0"/></c:when>
81 </c:choose>
82 <c:set var="sum_Number" value="${sum_Number + rs1.c_Number}" />
83 <c:set var="sum_Avg" value="${sum_Avg + (avg * rs1.c_Number)}" />
84 </c:forEach>
85 </table><br>
86 <table border="1" align="center">
87 <tr>
88 <td>총취득과목수</td>
89 <td>[${sno}] 과목</td>
90 </tr>
91 <tr>
92 <td>총취득학점수</td>
93 <td>[${sum_Number}] 학점</td>
94 </tr>
95 <tr>
96 <td>전체평균평점</td>
97 <td>[<fmt:formatNumber value="${sum_Avg div sum_Number}" pattern="0.00"/>] 점</td>
98 </tr>
99 </table>
100 </body>
101 </html>

```

평점 오류의 예

개인별 성적표

학과	학년	학번	성명
컴퓨터공학과	3	C0802	서희경

수번	과목번호	과목명	학점	등급
1	L0011	TOEIC연구	2	A+
2	L1031	SQL	3	A+
3	L1032	자바프로그래밍	3	A
4	L1041	컴퓨터네트워크	2	A+
5	L1042	Delphi	3	A+
6	L1051	웹서버관리	2	C+
7	L1052	전자상거래	3	B+

총취득과목수	[7] 과목
총취득학점수	[18] 학점
전체평균평점	[4.03] 점

개인별 성적표

학과	학년	학번	성명
컴퓨터공학과	3	C0802	서희경

수번	과목번호	과목명	학점	등급
1	L0011	TOEIC연구	2	A+
2	L1031	SQL	3	A+
3	L1032	자바프로그래밍	3	A
4	L1041	컴퓨터네트워크	2	A+
5	L1042	Delphi	3	A+
6	L1051	웹서버관리	2	C+
7	L1052	전자상거래	3	B+

총취득과목수	[7] 과목
총취득학점수	[18] 학점
전체평균평점	[4.25] 점

```

<c:choose>
  <c:when test= "${rs1.grade == "A+"}> <c:set var="avg" value="4.5"/></c:when>
  <c:when test= "${rs1.grade == "A"}> <c:set var="avg" value="4.0"/></c:when>
  <c:when test= "${rs1.grade == "B+"}> <c:set var="avg" value="3.5"/></c:when>
  <c:when test= "${rs1.grade == "B"}> <c:set var="avg" value="3.0"/></c:when>
  <c:when test= "${rs1.grade == "C+"}> <c:set var="avg" value="2.5"/></c:when>
  <c:when test= "${rs1.grade == "C"}> <c:set var="avg" value="2.0"/></c:when>
  <c:when test= "${rs1.grade == "D+"}> <c:set var="avg" value="1.5"/></c:when>
  <c:when test= "${rs1.grade == "D"}> <c:set var="avg" value="1.0"/></c:when>
  <c:when test= "${rs1.grade == "F"}> <c:set var="avg" value="0.0"/></c:when>
</c:choose>
  
```

[예제 13.16]

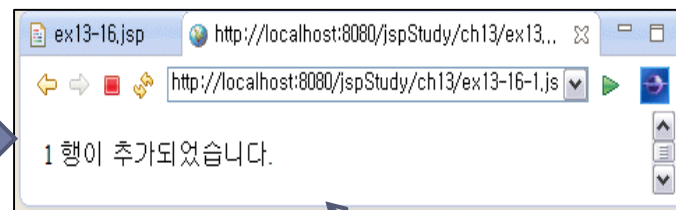
SG_Scores 테이블에 성적데이터(학번, 과목코드, 성적)을 입력하여 등급을 산출한 후 행을 추가하는 프로그램을 작성하시오.

ex13-16.jsp 테이블 행 추가 폼

http://localhost:8080/jspStudy/ch13/ex13-16.jsp

성적 입력 화면

학번	T1001
과목코드	L1032
성적	
<input type="button" value="입력"/> <input type="button" value="취 소"/>	



```
22 <body>
23 <center><h4> 성적 입력 화면 </h4>
24 <form method="post" action="ex13-16-1.jsp" name="scorefrm">
25 <table border="1" cellspacing="1">
26 <tr>
27 <td>학번 </td>
28 <td><input type="text" name="student_id"></td></tr>
29 <tr>
30 <td>과목코드</td>
31 <td><input type="text" name="course_id"></td></tr>
32 <tr>
33 <td>성적</td>
34 <td><input type="text" name="score"></td></tr>
35 <tr align="center">
36 <td colspan="2">
37 <input type="button" name="confirm" value="입력" OnClick="In_Check()">
38 <input type="reset" name="reset" value="취 소">
39 </td></tr>
40 </table>
41 </form></center>
42 </body>
43 </html>
```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
5
6 <c:set var="cgrade" value="F " />
7 <c:choose>
8     <c:when test='${param.score < 60}'><c:set var="cgrade" value="F " /></c:when>
9     <c:when test='${param.score < 65}'><c:set var="cgrade" value="D " /></c:when>
10    <c:when test='${param.score < 70}'><c:set var="cgrade" value="D+" /></c:when>
11    <c:when test='${param.score < 75}'><c:set var="cgrade" value="C " /></c:when>
12    <c:when test='${param.score < 80}'><c:set var="cgrade" value="C+" /></c:when>
13    <c:when test='${param.score < 85}'><c:set var="cgrade" value="B " /></c:when>
14    <c:when test='${param.score < 90}'><c:set var="cgrade" value="B+" /></c:when>
15    <c:when test='${param.score < 95}'><c:set var="cgrade" value="A " /></c:when>
16    <c:otherwise><c:set var="cgrade" value="A+" /></c:otherwise>
17 </c:choose>
18
19 <sql:update dataSource="jdbc/OracleDB" var="result">
20     INSERT INTO SG_Scores
21     (student_id, course_id, score, grade)
22     VALUES
23     (?, ?, ?, ?)
24     <sql:param value="${param.student_id}" />
25     <sql:param value="${param.course_id}" />
26     <sql:param value="${param.score}" />
27     <sql:param value="${cgrade}" />
28 </sql:update>
29 <c:out value="${result}" /> 행이 추가되었습니다.

```

context.xml 사용시
<sql:setDataSource> 생략

5. xml 라이브러리 (생략)

▶ xml 라이브러리

- ▶ 변수 설정, 조건 또는 반복처리, 문서 관련 태그

지시어	<%@ taglib prefix="x" url="http://java.sun.com/jsp/jstl/xml" %>
-----	---

기 능	태 그	설 명
변수설정	<x:set>	지정한 변수에 값을 저장
조건처리	<x:if>	<c:if>와 유사. 조건 판단 결과를 변수에 저장
	<x:choose>	<c:choose>와 유사, 조건문의 시작 태그
	<x:when>	조건 지정
	<x:otherwise>	<x:when> 조건이 아닐 때 처리
반복처리	<x:forEach>	반복처리
문서파싱	<x:parse>	xml 문서를 파싱
문서변환	<x:transform>	xml 문서를 XSL 스타일시트를 이용하여 문서 변형
값 전달	<x:param>	<x:transform> 태그 사이에 파라메타 값 전달

6. Functions 라이브러리

- ▶ Function 라이브러리
 - ▶ 문자열 처리에 관한 함수 제공
- ▶ 지시어
 - ▶ `<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>`
- ▶ 함수 목록
 - ▶ 주로 EL(Expression Language)식에서 사용
- ▶ 사용법
 - ▶ `${fn:함수명(인자1, 인자2, ...)}`
 - ▶ 예: `${fn:toLowerCase("HELLO")}` => hello

$\{fn: \text{함수명(인자1, 인자2, ...)}\}$

str, s1, s2, src, dest는 문자열을 의미

함 수 명	설 명
length(obj)	obj가 문자열이면 길이를 반환하고, 배열이나 List 나 Collection일 때 요소의 수를 반환
toLowerCase(str)	str을 소문자로 변환
toUpperCase(str)	str을 대문자로 변환
substring(str, idx1, idx2)	str에서 idx1부터 idx2까지 문자열 추출
substringAfter(s1, s2)	s1에서 s2 이후의 문자열을 반환
substringBefore(s1, s2)	s1에서 s2 이전의 문자열을 반환
trim(str)	str의 공백 제거
replace(str, src, dest)	str에서 src 문자열을 dest 문자열로 치환
indexOf(s1, s2)	s1에서 s2의 시작 인덱스 반환
startsWith(s1, s2)	s1이 s2로 시작하면 true, 그렇지 않으면 false 반환
endsWith(s1, s2)	s1이 s2로 끝나면 true, 그렇지 않으면 false 반환
contains(s1, s2)	s1이 s2를 포함하면 true, 그렇지 않으면 false 반환
containsIgnoreCase(s1, s2)	s1이 s2를 포함하면 true, 그렇지 않으면 false 반환. 대소문자 구분 안함
split(s1,s2)	s2를 기준으로 s1을 분리하여 배열로 반환
join(arr,str)	arr 배열의 모든 항목을 str을 합쳐 반환
escapeXml(str)	XML 객체 참조에 해당하는 특수문자 처리

【예제 13.18】

"It is time to take a break." 문자열 변수를 선언하고, 문자열의 길이, "to"의 위치, "is"를 "was"로 변경, 문자열을 대문자로 변경하는 프로그램을 작성하시오.

JSTL Function 사용 예

str?	It is time to take a break.
str의 길이 ?	27
str의 to 위치?	11
is를 was로 변경?	It was time to take a break.
str를 대문자로 변환?	IT IS TIME TO TAKE A BREAK.

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8" %>
3  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4  <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
5
6  <c:set var="str" value="It is time to take a break." />
7  <table border="1">
8      <tr>
9          <th colspan="2" align="center">JSTL Function 사용 예 </th>
10     </tr>
11     <tr>
12         <td>str?</td>
13         <td>${str}</td>
14     </tr>
15     <tr>
16         <td>str의 길이 ?</td>
17         <td>${fn:length(str)}</td>
18     </tr>
19     <tr>
20         <td>str의 to 위치?</td>
21         <td>${fn:indexOf(str, "to")}</td>
22     </tr>
23     <tr>
24         <td>is를 was로 변경?</td>
25         <td>${fn:replace(str, "is", "was")}</td>
26     </tr>
27     <tr>
28         <td>str를 대문자로 변환?</td>
29         <td>${fn:toUpperCase(str)}</td>
30     </tr>
31 </table>

```

【예제 13.19】 "It is time to take a break." 문자열 변수를 선언하고, "time" 문자열의 포함여부, "7부터 10까지 문자열 추출, "time" 문자열의 이전/이후 문자열, "It"로 문자열의 시작 여부의 결과를 출력하시오.

ex13-19.jsp http://localhost:8080/jspStudy/ch13/ex13...

http://localhost:8080/jspStudy/ch13/ex13-19.jsp

str?	It is time to take a break.
time 문자열 포함 여부?	true
time 추출?	ime
time 이후의 문자열 반환?	to take a break.
time 이전의 문자열 반환?	It is
It로 문자열 시작 여부?	true

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" %>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
5
6 <c:set var="str" value="It is time to take a break." />
7 <table border="1">
8   <tr>
9     <th colspan="2" align="center">JSTL Function 사용 예[2] </th>
10  </tr>
11  <tr>
12    <td>str?</td>
13    <td>${str}</td>
14  </tr>
15  <tr>
16    <td>time 문자열 포함 여부?</td>
17    <td>${fn:contains(str, "time")}</td>
18  </tr>
19  <tr>
20    <td>time 추출?</td>
21    <td>${fn:substring(str,7,10)}</td>
22  </tr>
23  <tr>
24    <td>time 이후의 문자열 반환?</td>
25    <td>${fn:substringAfter(str, "time")}</td>
26  </tr>
27  <tr>
28    <td>time 이전의 문자열 반환?</td>
29    <td>${fn:substringBefore(str, "time")}</td>
30  </tr>
31  <tr>
32    <td>It로 문자열 시작 여부?</td>
33    <td>${fn:startsWith(str, "It")}</td>
34  </tr>
35 </table>

```

강 의 내 용 요약 정 리

▶ JSTL?

- ▶ JSP Standard Tag Library) 약어. JSP 표준 태그 라이브러리
- ▶ 라이브러리란 여러 프로그램이 공통으로 사용하는 코드를 모아놓은 코드 집합
- ▶ core, format, database, xml, functions 5개의 라이브러리
- ▶ functions 라이브러리는 함수, 4 개의 라이브러리들은 커스텀 액션 태그

라이브러리	기 능 설 명
core	● 변수 선언. 흐름의 제어, 다른 JSP 페이지로 제어 이동 기능
format	● 숫자, 날짜, 시간을 형식 지정과 국제화, 다국어 지원 기능
database	● 데이터베이스의 데이터 입력, 수정, 삭제, 조회 기능
xml	● xml 문서 처리 기능
functions	● 문자열 처리 함수 기능

강 의 내 용 요약 정 리

JSP 페이지 상단에 코딩

라이브러리	JSTL 접두어와 지시어
core	<%@ taglib prefix="c" url="http://java.sun.com/jsp/jstl/core" %>
format	<%@ taglib prefix="fmt" url="http://java.sun.com/jsp/jstl/fmt" %>
database	<%@ taglib prefix="sql" url="http://java.sun.com/jsp/jstl/sql" %>
xml	<%@ taglib prefix="x" url="http://java.sun.com/jsp/jstl/xml" %>
functions	<%@ taglib prefix="fn" url="http://java.sun.com/jsp/jstl/functions" %>

- JSTL 1.1 라이브러리를 구성하는 파일
 - **jstl.jar**, **standard.jar**
- 저장위치
 - **WEB-INF/lib**

기본문법 끝



다음장 주제

- ▶ 응용 프로그래밍
 - ▶ 14장 회원관리 프로그램
 - ▶ 15장 페이지 중심 설계의 자유게시판 제작
 - ▶ 16장 MVC 설계 패턴
- ▶ 실무 프로그래밍
 - ▶ 17장 스프링 프레임워크
 - ▶ 18장 마이바티스와 마이바티스-스프링
 - ▶ 19장 스프링 MVC 자유게시판 제작



장 주제

14장. 회원관리프로그램

