



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음

Chapter 11. JSTL의 이해와 활용

목차

1. JSTL의 개념과 구성
2. 핵심 라이브러리의 주요 태그
3. [응용실습]스크립트릿을 JSTL로 변환

01. JSTL의 개념과 구성

1. JSTL이란?

- JSTL(JSP Standard Tag Library)은 커스텀 태그 라이브러리 기술을 이용해서 일반적으로 필요한 기능들을 표준화한 것으로 크게 핵심(CORE), xml, I18N(국제화), 데이터베이스(SQL), 함수(functions) 라이브러리로 구성된다.
- 커스텀 태그 기반이므로 JSTL을 사용하는 방법은 일반적인 커스텀 태그와 같다.
- 다운로드 : <http://tomcat.apache.org/taglibs/standard/>
- **taglib 지시어 사용법(core 라이브러리의 경우)**

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

[표 11-1] JSTL 라이브러리별 URI 및 prefix

라이브러리	URI	prefix
핵심	http://java.sun.com/jsp/jstl/core	c
XML	http://java.sun.com/jsp/jstl/xml	x
I18N	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

01. JSTL의 개념과 구성

2. JSTL 구성

- JSTL은 태그가 제공하는 기능의 성격에 따라 5가지 주요 라이브러리로 구분된다.
- 교재에서는 Core(핵심) 을 중심으로 살펴본다.

[표 11-2] JSTL 라이브러리

라이브러리	기능	태그	접두어
핵심	General Purpose Actions	• catc • out • remove • set	c
	Conditional Actions	• choose • when • otherwise • if	
	Iterator Actions	• forEach • forTokens	
	URL Related Actions	• import • redirect • url • param	
XML	Core	• out • parse • set	x
	Flow Control	• choose • when • otherwise • forEach • if	
	Transformation	• transform • param	

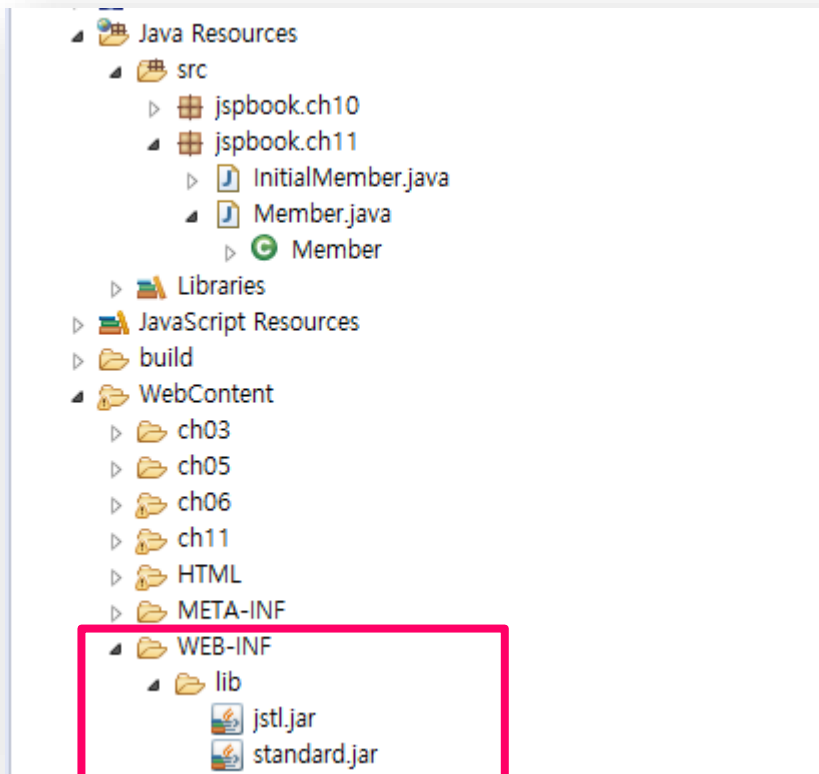
01. JSTL의 개념과 구성

- I18N 은 다국어 처리와 관련된 기능을 제공하고 sql(데이터베이스)는 간단하게 데이터베이스 관련 작업을 지원하며 functions(함수)는 여러 부가기능들을 제공한다.

라이브러리	기능	태그	접두어
I18N	Locale	• setLocale	fmt
	Message Formatting	• bundle • message • param • setBundle • requestEncoding	
	Number and DateFormatting	• formatNumber • formatDate • parseDate • parseNumber • setTimeZone • timeZone	
데이터베이스	Database Access	• setDataSource • query • dateParam • param • transaction • update • dateParam • param	sql
함수		• ontains • containsIgnoreCase • endsWith • escapeXml • indexOf • join • length • replace • split • startsWithsubstring • substringAfter • substringBefore • toLowerCase • toUpperCase • trim	fn

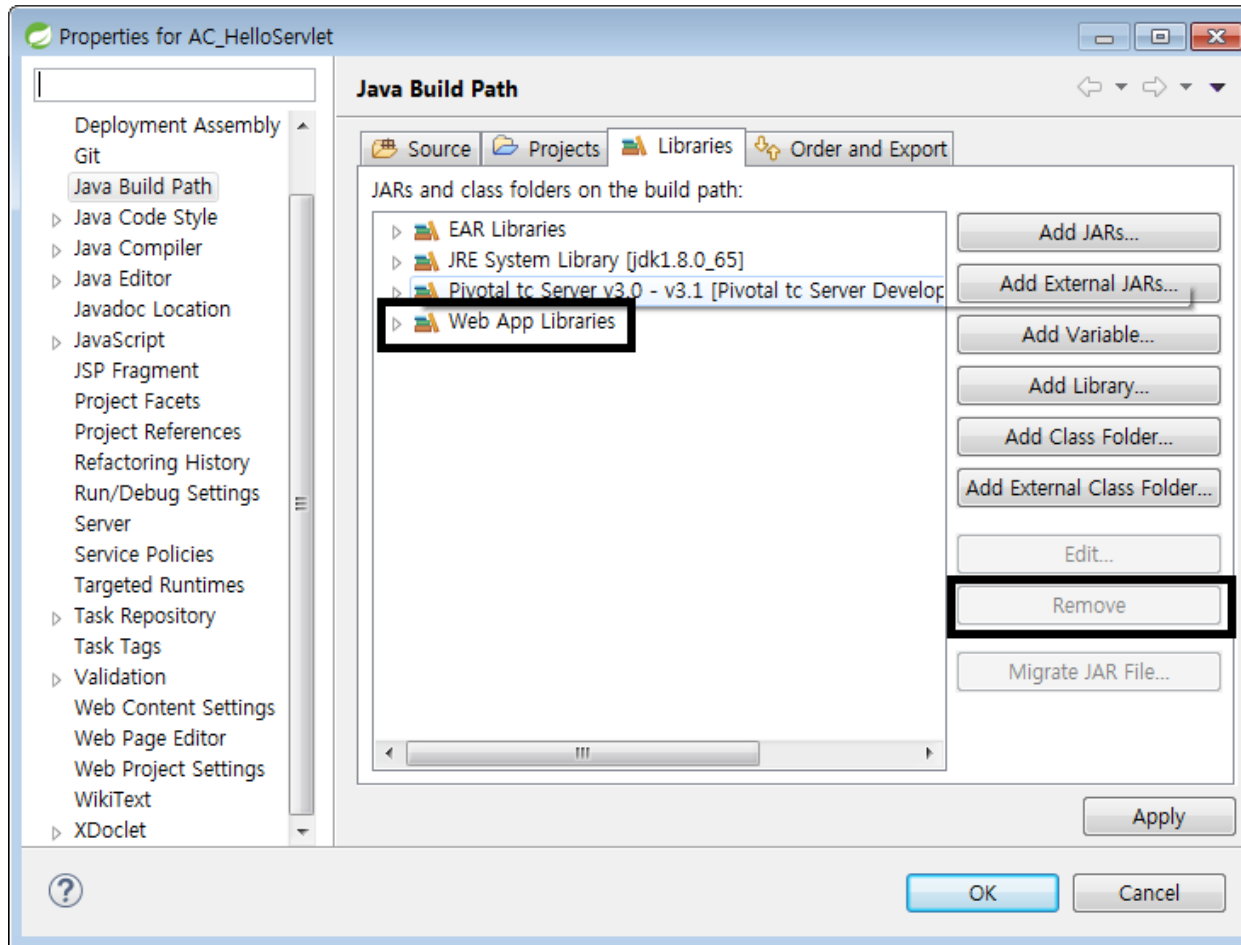
JSTL 설치 (1/3)

- 이클래스 강의자료 게시판에서 JSTL 라이브러리 파일을 다운로드
- 압축을 풀고 [lib] 폴더에서 jstl.jar와 standard.jar 파일을 찾는다
- 이클립스 프로젝트의 [WebContent] → [WEB-INF] → [lib] 폴더로 복사한다.



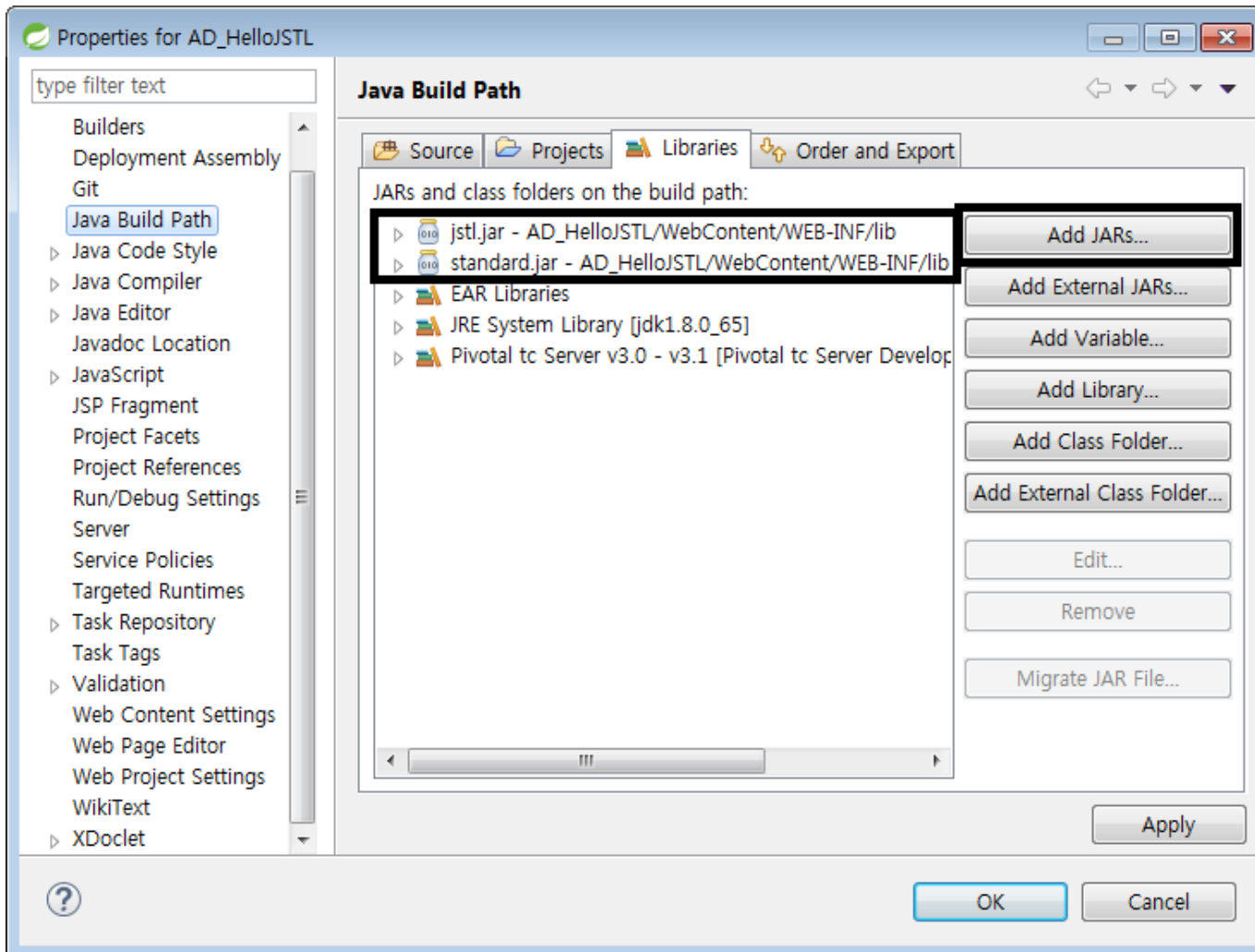
JSTL 설치 (2/3)

- 프로젝트 우클릭 → Properties → Java Build Path



JSTL 설치 (3/3)

- WebContent/WEB-INF/lib에 복사한 2개의 jar 파일을 추가



01. JSTL의 개념과 구성 (Member.java)

```
package jspbook.ch11;

public class Member {
    private String name;
    private String email;

    public Member(String name, String email) {
        this.name = name;
        this.email = email;
    }

    public Member() {
        name= " 홍길동";
        email="test@test.net";
    }

    public String getName() { return name; }
    public String getEmail() { return email; }


    public void setEmail(String email)      {      this.email = email;      }
    public void setName(String name)        {      this.name = name;      }

    public String toString() {
        return (name + "/" + email);
    }
}
```

01. JSTL의 개념과 구성 (initMembers.jsp)

```
page_control... Member.java initialMemb... printMember... Member.java
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="jspbook.ch11.*, java.util.ArrayList" %>
4
5 <html>
6 <head><title>Initialize member data</title></head>
7 <body>
8 <%
9   ArrayList<Member> members = new ArrayList<Member>();
10
11   for (int i = 0; i < 8; i++) {
12     Member data = new Member("황희정" + i, "test" + i + "@test.net");
13     members.add(data);
14   }
15
16   members.add(new Member("홍길동", null));
17   members.add(new Member("김길동", null));
18
19   application.setAttribute("members", members);
20   application.setAttribute("member", new Member());
21
22   response.sendRedirect("printMembers.jsp");
23 %>
24 </body>
25 </html>
26
```

01. JSTL의 개념과 구성



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="jspbook.ch11.*, java.util.ArrayList" %>
4
5 <html>
6 <head><title>Print Members</title></head>
7 <body>
8 <%
9     ArrayList<Member> members = (ArrayList<Member>)application.getAttribute("members");
10
11     for (Member m: members) {
12         out.println(m.toString() + "<br>");
13     }
14     out.println("<br>");
15
16     Member lastMember = (Member) application.getAttribute("member");
17     out.println(lastMember.toString());
18 %>
19 </body>
20 </html>
21 |
```

실행결과

```
황희정0/test0@test.net  
황희정1/test1@test.net  
황희정2/test2@test.net  
황희정3/test3@test.net  
황희정4/test4@test.net  
황희정5/test5@test.net  
황희정6/test6@test.net  
황희정7/test7@test.net  
홍길동/null  
김길동/null  
  
홍길동/test@test.net
```

속성이름: members

속성이름: member

02. 핵심 라이브러리의 주요 태그

3. 순환 처리 태그

■ <c:forEach> 태그 – 교재 p. 490

- 반복문과 관련된 태그로 자바의 for 문과 유사하다. 가장 중요하고 널리 쓰이는 JSTL 태그 중 하나임.
- 자바 프로그래밍 언어와 달리 중첩 반복문은 사용할 수 없음.
- 사용법
 - 컬렉션 객체의 크기만큼 반복

```
<c:forEach var="varName" items="collection" >  
body content  
</c:forEach>
```

- 지정된 횟수 반복

```
<c:forEach [varStatus="varStatusName"] begin="begin" end="end" [step="step"]>  
body content  
</c:forEach>
```

02. 핵심 라이브러리의 주요 태그

- varStatus는 반복 과정에서 프로그램적으로 필요한 여러 정보를 제공함.
- 자바 언어에서의 for와 같은 자유도는 없기 때문에 모든 반복 처리에 <c:forEach>를 사용하기는 어렵고 데이터 생성시 <c:forEach>에서 처리되기 편리한 형태로 가공해주는 것이 필요하다.

[표 11-9] <c:forEach> 속성 값

속성	필수	기본 값	설명
items	N	없음	반복을 위한 데이터를 가진 아이템의 컬렉션
begin	N	0	반복 시작 번호
end	N	컬렉션의 마지막 값	반복 끝 번호
step	N	1	반복의 증가분
var	N	없음	현재 아이템이 있는 변수
varStatus	N	없음	반복 상태 값이 있는 변수

예제 : forEach

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl" %>
3 <html>
4 <head>
5   <title>ch11 : forEach</title>
6 </head>
7 <body bgcolor="#FFFFFF">
8
9   <h3>ForEach 방식</h3>
10  <c:forEach var="m" items="${members}">
11    name: ${m.name}
12    email: ${m.email} <BR>
13    <HR>
14  </c:forEach>
15
16  <h3>전형적인 For 방식</h3>
17  <c:forEach varStatus="i" begin="0" end="5" step="2">
18    index: ${i.index}
19    count: ${i.count} <BR>
20    <HR>
21  </c:forEach>
22
23 </body>
24 </html>
25
```

ForEach 방식

name: 황희정0 email: test0@test.net

name: 황희정1 email: test1@test.net

name: 황희정2 email: test2@test.net

name: 황희정3 email: test3@test.net

name: 황희정4 email: test4@test.net

name: 황희정5 email: test5@test.net

name: 황희정6 email: test6@test.net

name: 황희정7 email: test7@test.net

name: 홍길동 email:

name: 김길동 email:

전형적인 For 방식

index: 0 count: 1

index: 2 count: 2

index: 4 count: 3

02. 핵심 라이브러리의 주요 태그

1. 기본 기능 태그

■ <c:out> 태그 – 교재 p. 478

- 초기에는 **jsp 표현식(<%= %>)**을 대체하기 위해 개발되었으나 표현언어가 JSP에 기본으로 제공 되면서 사용 빈도는 줄었으나 몇몇 옵션은 유용하게 사용할 수 있다.

▪ 사용법

- 태그 바디가 없는 경우

```
<c:out value=" value " [escapeXml=" {true | false} " ] [default=" defaultValue "] />
```

- 태그 바디가 있는 경우

```
<c:out value=" value " [escapeXml=" {true | false} " ]>
```

default value (value에 내용이 없을 때 출력될 기본 값)

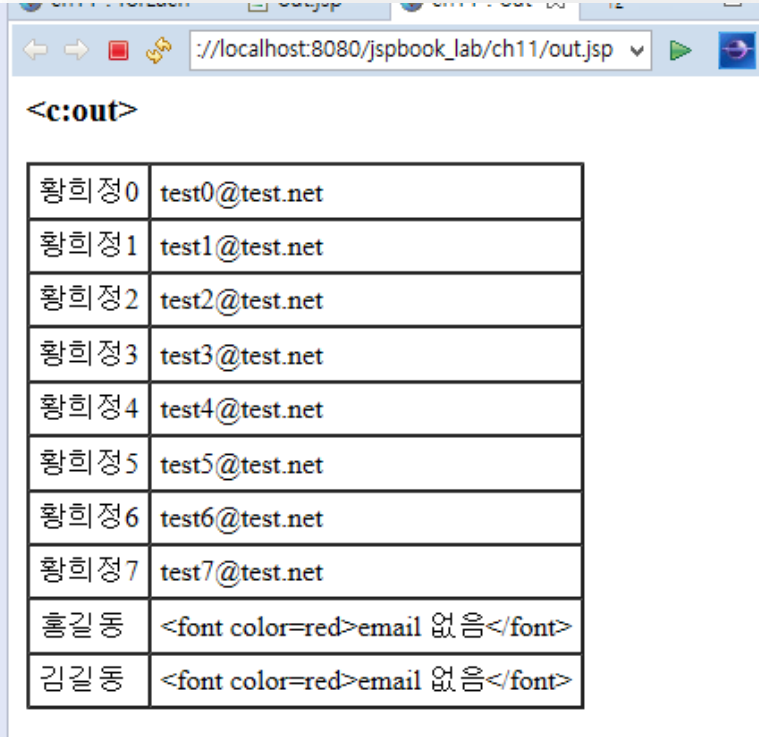
```
</c:out>
```

[표 11-3] <c:out> 태그 속성 값

속성	필수	기본 값	설명
value	Y	없음	출력될 내용 또는 표현식이다.
default	N	태그 바디에 있는 내용	value 값에 내용이 없는 경우에 출력할 내용으로, 태그 바디 혹은 속성 값 형태로 올 수 있다.
escapeXml	N	true	출력될 내용에 <, >, &, ' 등 문자를 일반 문자로 변환할 것인지 결정한다. 예를 들어 출력될 내용에 HTML 태그가 포함되어 있다면 이 값을 false로 해야 태그가 반영된 내용이 화면에 보인다. 만일 true로 할 경우 태그가 그대로 화면에 보이게 된다.

02. 핵심 라이브러리의 주요 태그

```
11 <table border="1" cellpadding="5" cellspacing="0">
12   <c:forEach var="member" items="${members}">
13     <tr>
14       <td> <c:out value="${member.name}"/> </td>
15       <td> <c:out value="${member.email}" escapeXml=" true">
16         <font color=red>email 없음 </font>
17       </c:out>
18     </td>
19   </tr>
20 </c:forEach>
21 </table>
```

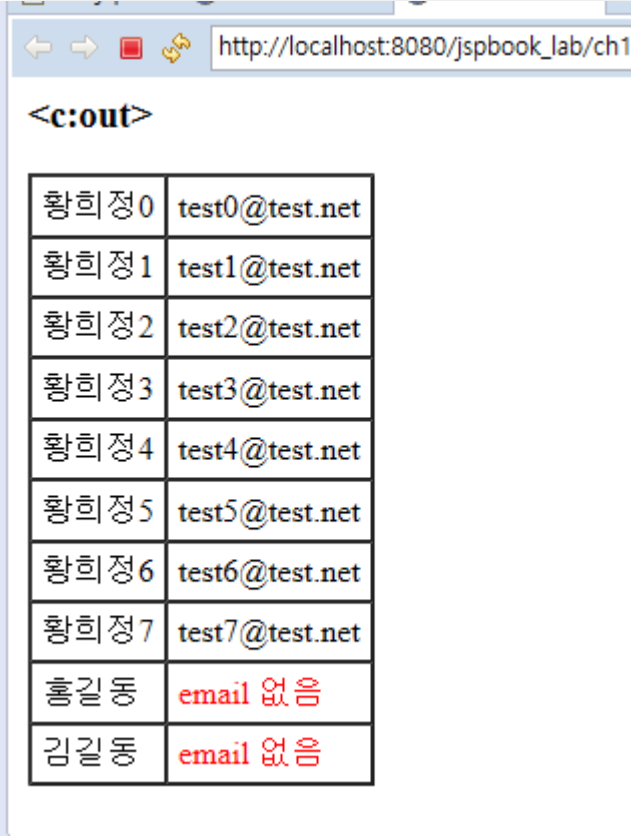


<c:out>

황희정0	test0@test.net
황희정1	test1@test.net
황희정2	test2@test.net
황희정3	test3@test.net
황희정4	test4@test.net
황희정5	test5@test.net
황희정6	test6@test.net
황희정7	test7@test.net
홍길동	email 없음
김길동	email 없음

02. 핵심 라이브러리의 주요 태그

```
11 <table border="1" cellpadding="5" cellspacing="0">
12   <c:forEach var="member" items="${members}">
13     <tr>
14       <td> <c:out value="${member.name}"/> </td>
15       <td> <c:out value="${member.email}" escapeXml="false">
16         <font color=red>email 없음 </font>
17       </c:out>
18     </td>
19   </tr>
20 </c:forEach>
21 </table>
```



<c:out>

황희정0	test0@test.net
황희정1	test1@test.net
황희정2	test2@test.net
황희정3	test3@test.net
황희정4	test4@test.net
황희정5	test5@test.net
황희정6	test6@test.net
황희정7	test7@test.net
홍길동	email 없음
김길동	email 없음

02. 핵심 라이브러리의 주요 태그

■ <c:set> 태그

- <c:set> 태그는 변수 값을 설정하거나 객체의 멤버변수 값을 설정할 때 사용한다.

- 사용법

- 해당 범위에 속성 값을 추가하는 경우 (바디가 올 수도 있다)

```
<c:set value=" value " var=" varName " [scope="{ page | request | session | application } "]/>
```

- 특정 target 객체에 새로운 속성 값을 설정하는 경우 (바디가 올 수도 있다)

```
<c:set target=" target " property=" propertyName " value=" value " />
```

[표 11-4] <c:set> 태그 속성 값

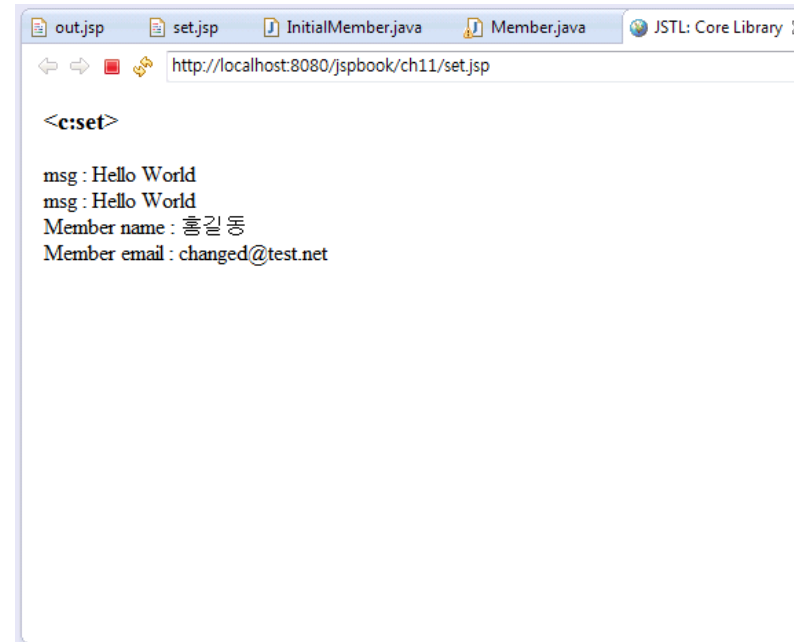
속성	필수	기본 값	설명
value	N	없음	저장할 변수 값
target	N	없음	값이 저장될 객체 이름
property	N	없음	target 객체의 멤버변수 이름
var	N	없음	값이 저장될 변수 이름
scope	N	page	값이 저장될 범위(page, session, request, application)

02. 핵심 라이브러리의 주요 태그

- [실습] set.jsp – 교재 p.481 참고

```
09 <h3>&lt;c:set&gt;</h3>
10 <c:set value="Hello World" var="msg" />
11 msg : ${msg} <BR>
12 msg : <%=pageContext.getAttribute("msg") %> <BR>
13
14 <c:set target="${member}" property="email" value="changed@test.net" />
15 Member name : ${member.name} <BR>
16 Member email : ${member.email}
```

page 내장 객체에 속성을 관리하기 위해서는
pageContext.setAttribute(...) &
pageContext.getAttribute() 메소드를 사용



[그림 11-3] set.jsp 실행 결과

02. 핵심 라이브러리의 주요 태그

2. 조건 처리 태그

■ <c:if> 태그

- <c:if> 태그는 조건에 따라 바디 내용을 처리 한다.
- 자바의 if와 비슷하지만 else문은 지원하지 않는다 (단순 조건 체크만 가능)
- 자바 프로그래밍 언어와 달리 중첩된 조건문은 사용할 수 없다.
- **사용법**

- 바디 내용이 없는 경우

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

- 바디 내용이 있는 경우

```
<c:if test="testCondition" [var="varName"] [scope="{ page | request | session | application }"]>
```

Body content

```
</c:if>
```

[표 11-7] <c:if> 태그 속성 값

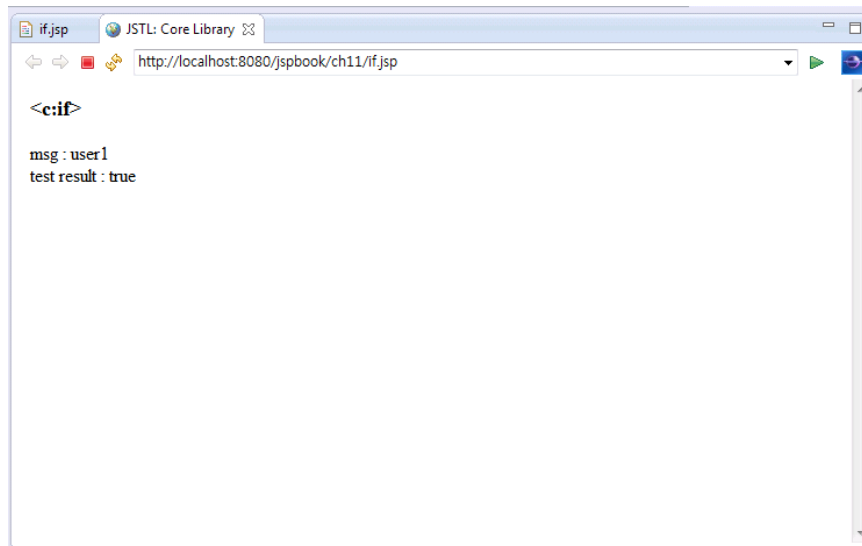
속성	필수	기본 값	설명
test	Y	없음	검사할 조건
var	N	없음	test 조건의 결과를 저장할 변수(결과는 true 혹은 false)
scope	N	page	변수가 저장될 범위

02. 핵심 라이브러리의 주요 태그

- [실습] if.jsp – [교재 p.486 참고](#)

- <c:set>을 이용해 값을 설정 하고 조건 체크를 통해 <c:if> 동작을 확인

```
09 <c:set value="user1" var="msg" />
10 msg : ${msg}" <BR>
11
12 <c:if test="${msg == 'user1'}" var="result">
13 Test result : ${result}
14 </c:if>
```

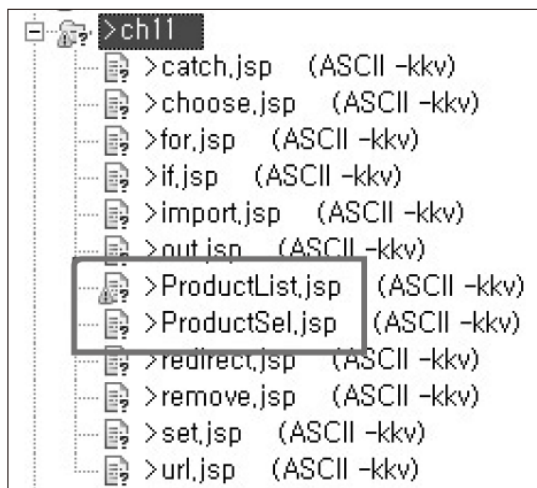


[그림 11-7] if.jsp 실행 결과

03. [응용실습]스크립트릿을 JSTL로 변환

1. 실습 개요

- 10장에서 만들었던 ProductList.jsp, ProductSel.jsp 소스를 JSTL 버전으로 변경 함.
- 스크립트릿 대신 표현언어와 JSTL 을 사용하는 JSP 개발을 위한 실전 응용 실습.
- 준비작업
 - 기존 10장의 jsp 소스를 ch11로 복사한다.
 - 자바 빈즈 클래스는 기존의 패키지를 그대로 사용할 것이므로 따로 복사하지 않아도 된다.



[그림 11-13] 파일 준비

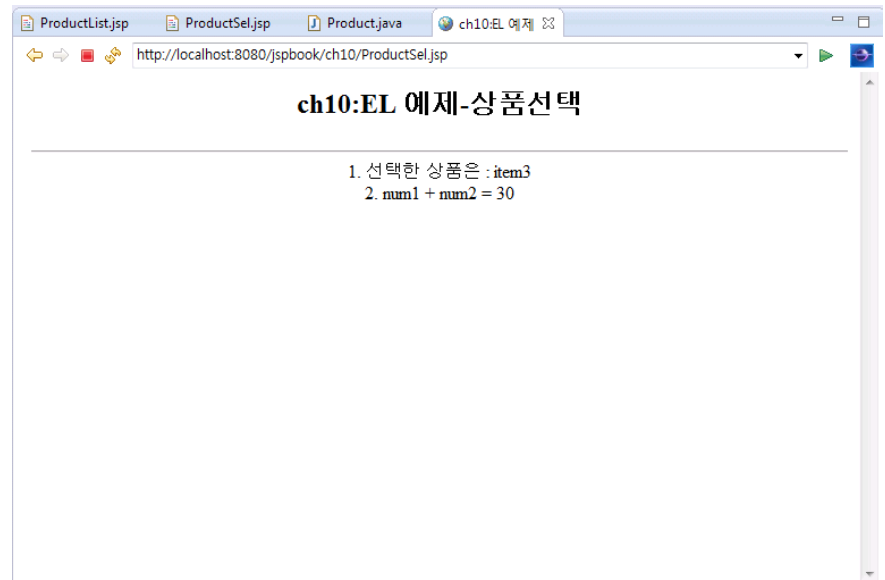
03. [응용실습]스크립트릿을 JSTL로 변환

2. 소스 작성

- ProductList.jsp - **교재 p.501 ~ 502 참고**
 - ProductSel.jsp 는 수정사항 없음.
 - 실행 결과는 10장에서와 동일함.



[그림 10-1] ProductList.jsp 실행 화면



[그림 10-2] ProductSel.jsp 실행 화면

02. [기본실습] 표현 언어의 기본 이해 (p. 431-432)

■ 주요 소스코드 분석(Product.java) – 교재 p. 431-432

- 데이터 관리를 위한 빈즈 클래스
- 상품정보는 편의상 배열로 만들어 제공
- 표현언어 실습을 위해 num1, num2 라는 변수를 미리 설정해 둠.

```
1 package jspbook.ch10;
2
3 public class Product {
4
5     private String[] productList = {"item1","item2","item3","item4","item5"};
6
7     private int num1 = 10;
8     private int num2 = 20;
9
10    public int getNum1() {
11        return num1;
12    }
13
14    public int getNum2() {
15        return num2;
16    }
17
18    public String[] getProductList() {
19        return productList;
20    }
21 }
```

02. [기본실습] 표현 언어의 기본 이해

■ 주요 소스코드 분석(ProductList.jsp) – 교재 p. 432

- 상품 목록을 HTML 의 <select><option> 을 이용해 출력하는 JSP
- <jsp:useBean>을 이용해 Product 클래스 사용
- for 문을 이용해 배열 데이터를 <select><option> 문과 조합해 출력

```
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>ch10:EL 예제</title>
8 </head>
9 <body>
10 <div align="center">
11 <H2>ch10:EL 예제-상품목록</H2>
12 <HR>
13 <form name=form1 method=POST action=ProductSel.jsp>
14   <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>
15   <select name="sel">
16     <%
17       for(String item : product.getProductList()) {
18         out.println("<option>" + item + "</option>");
19       }
20     %>
21   </select>
22   <input type="submit" value="선택"/>
23 </form>
24 </div>
25 </body>
26 </html>
27
```

02. [기본실습] 표현 언어의 기본 이해

■ 주요 소스코드 분석(ProductSel.jsp) – 교재 p. 433

- 선택된 상품정보를 출력하는 JSP
- 표현언어를 통해 선택된 상품과 product 객체의 정보를 출력

```
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>ch10:EL 예제</title>
8 </head>
9 <body>
10 <div align="center">
11 <H2>ch10:EL 예제-상품선택</H2>
12 <HR>
13 1. 선택한 상품은 : ${param.sel} <BR>
14 2. num1 + num2 = ${product.num1+product.num2} <BR>
15 </div>
16 </body>
17 </html>
18
```

02. [기본실습] 표현 언어의 기본 이해

```
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>ch10:EL 예제</title>
8 </head>
9 <body>
10 <div align="center">
11 <H2>ch10:EL 예제-상품목록</H2>
12 <HR>
13 <form name=form1 method=POST action=ProductSel.jsp>
14   <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>
15   <select name="sel">
16     <%
17       for(String item : product.getProductList()) {
18         out.println("<option>" + item + "</option>");
19       }
20     %>
21   </select>
22   <input type="submit" value="선택"/>
23 </form>
24 </div>
25 </body>
26 </html>
```

```
03 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
....
13 <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>
14 <select name="sel">
15   <c:forEach items="${product.productList}" var="item">
16     <option>${item}</option>
17   </c:forEach>
18 </select>
```



프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음