



표준태그라이브러리 (JSTL:JSP Standard Tag Library)

Contents

- ❖ 1. JSTL 기초
- ❖ 2. JSTL core 태그
- ❖ 3. JSTL formatting 태그

1. 표준 태그 라이브러리의 기초

❖ 표준 태그 라이브러리(JSTL)란?

- JSTL – JSP Standard Tag Library의 약자
- JSP에서 개발자가 추가할 수 있는 커스텀 태그를 모아놓은 표준
- **커스텀 태그**
 - <jsp:include> 와 같은 태그를 개발자가 추가할 수 있는 태그
- JSP 페이지에서 많이 사용되는 논리적인 판단, 반복 처리, 포맷 처리를 위한 커스텀 태그를 표준으로 만들어서 정의한 것
=> JSTL(JSP Standard Tag Library)

1. 표준 태그 라이브러리의 기초

❖ 표준 태그 라이브러리(JSTL)의 역할

- JSTL을 사용하면 스크립틀릿 코드의 사용을 줄여줌
- 더욱 간결하고 이해하기 쉬운 JSP 코드 작성이 가능
- MVC 패턴에서 View(표현) 역할 강화
- EL과 함께 사용되어 코드 가독성 향상

1. 표준 태그 라이브러리의 기초

❖ JSTL이 제공하는 태그의 종류

접두어 – JSP 페이지가 커스텀
태그를 호출할 때 사용

라이브러리	주요 기능	접두어	관련 URI
core	변수지원 흐름제어 URL 처리	c	http://java.sun.com/jsp/jstl/core
functions	컬렉션 처리 문자열(String) 처리	fn	http://java.sun.com/jsp/jstl/functions
formatting	날짜/숫자/통화 포맷	fmt	http://java.sun.com/jsp/jstl/fmt
SQL	DB 접근(보안상 비추천)	sql	http://java.sun.com/jsp/jstl/sql
XML	XML 처리	x	http://java.sun.com/jsp/jstl/xml

관련 URI – JSTL이 제공하는
커스텀 태그를 구분해 주는 식별자

1. 표준 태그 라이브러리의 기초

❖ JSTL 라이브러리 다운로드

- JSTL 을 사용하려면 JSTL 1.2 버전을 구현한 jar 파일을 다운로드해야 한다.
- 아래 파일명으로 검색해서 다운로드
 - javax.servlet.jsp.jstl1.2.1.jar
https://jar-download.com/?search_box=jstl-2.1
 - javax.servlet.jsp.jstl1-api-1.2.1.jar
https://jar-download.com/?search_box=javax.servlet.jsp.jstl--1.2.1
- 다운로드한 jar 파일을 아래 폴더에 복사
 - 프로젝트명 > WebContent > WEB-INF > lib > 아래 복사

1. 표준 태그 라이브러리의 기초

❖ JSTL 라이브러리 다운로드

- JSTL 을 사용하려면 JSTL 1.2 버전을 구현한
- 아래 파일명으로 검색해서 다운로드

- javax.servlet.jsp.jstl1.2.1.jar

https://jar-download.com/?search_box=jstl-2.1

- javax.servlet.jsp.jstl1-api-1.2.1.jar

https://jar-download.com/?search_box=javax.servlet.jsp.jstl--1.2.1

Tomcat 9.0에서만 가능

- 다운로드한 jar 파일을 아래 폴더에 복사

- 프로젝트명 > WebContent > WEB-INF > lib > 아래 복사

1. 표준 태그 라이브러리의 기초

❖ JSTL 라이브러리 다운로드

- JSTL 을 사용하려면 JSTL 1.2 버전을 구현한
- 아래 파일명으로 검색해서 다운로드

- **javax.servlet.jsp.jstl1.2.1.jar**

https://jar-download.com/?search_box=jstl-2.1

Tomcat 9.0에서만 가능

- **javax.servlet.jsp.jstl1-api-1.2.1.jar**

https://jar-download.com/?search_box=jstl-2.1

<https://tomcat.apache.org>

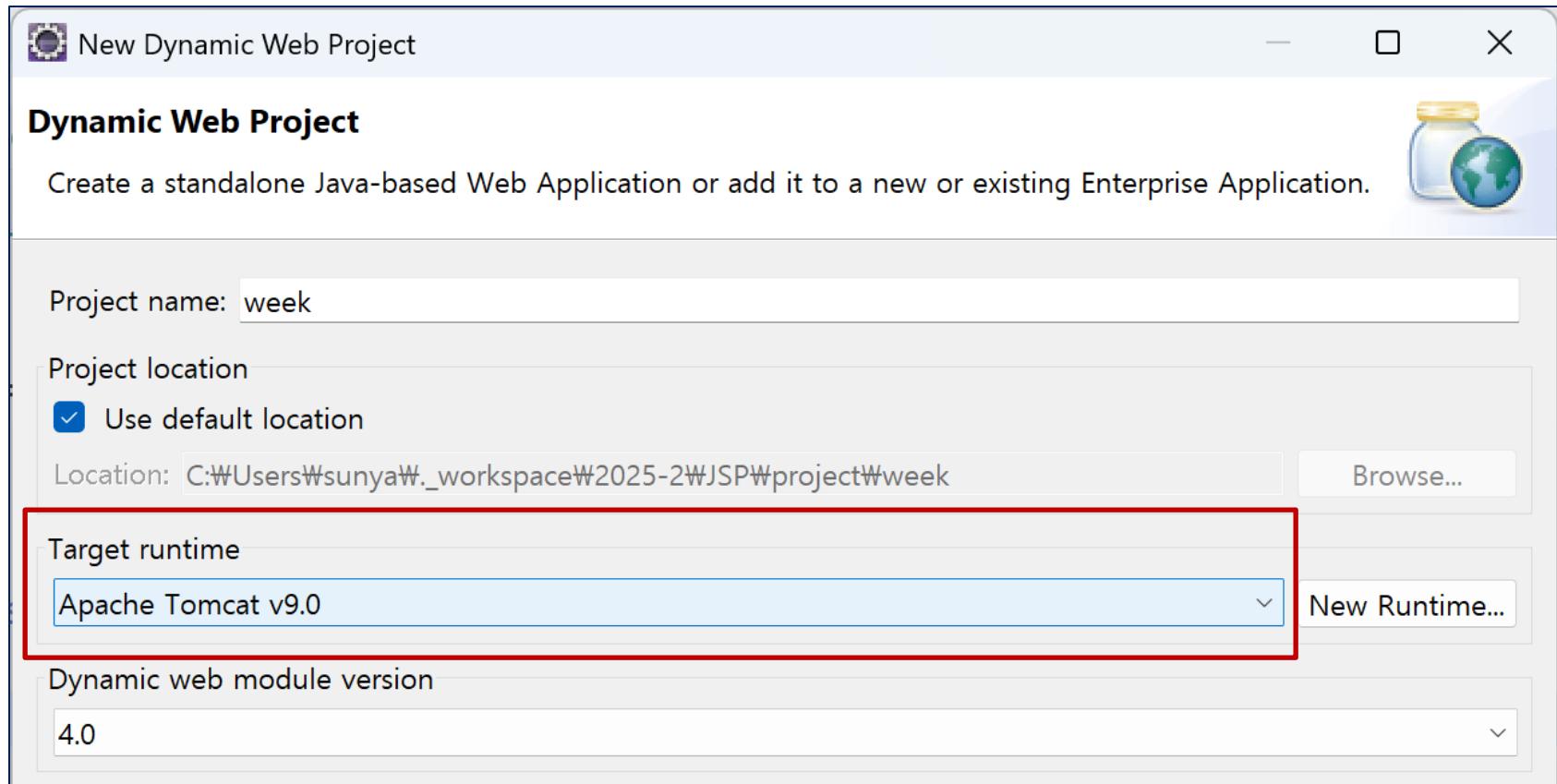
tomcat 9.0 다운로드 후 설치

- 다운로드한 jar 파일을 아래 폴더에 복사

- **프로젝트명 > WebContent > WEB-INF > lib > 아래 복사**

1. 표준 태그 라이브러리의 기초

- ❖ Dynamic Web Project 생성 시
 - Target runtime 항목 – Apache Tomcat v9.0 선택

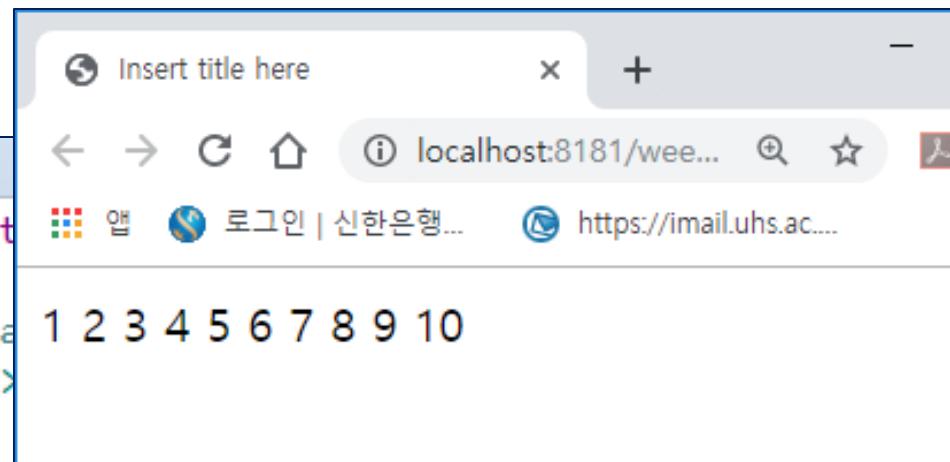


1. 표준 태그 라이브러리의 기초

❖ JSP 기본 예제

▪ if-else 블록과 for문 사용 예제

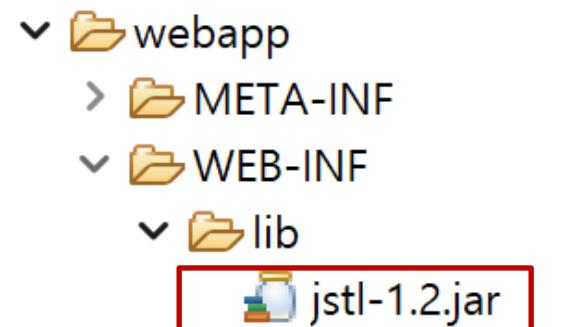
```
ifElse.jsp ✎ JSTL.jsp
1 <%@ page language="java" content-
2     pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head><meta
4 <title>Insert title here</title>
5<body>
6<%
7     int[] list = {1,2,3,4,5,6,7,8,9,10};
8
9     if (list.length > 0){
10         for (int i : list)
11             out.println(i + " ");
12     } else {
13         out.println("배열에 저장된 데이터가 없습니다.");
14     }
15 %>
16
17<
18
19 </body></html>
```



1. 표준 태그 라이브러리의 기초

❖ JSTL을 사용한 예제

▪ if-else 블록과 for문 사용 예제



ifElse.jsp JSTL.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 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6 <body>
7 <%
8     int[] list = {1,2,3,4,5,6,7,8,9,10};
9 %>
10 <c:if test="<%= list.length > 0 %>">
11     <c:forEach var="data" items="<%list %>">
12         ${data }
13     </c:forEach>
14 </c:if>
15 <c:if test="<%= list.length == 0 %>">
16     데이터가 없습니다.
17 </c:if>
18 </body></html>
```

-HTML 태그와 비슷한 태그를 사용하여
보기 좋고 이해하기 쉬운 코드로 작성 가능
-스크립틀릿 코드 영역이 줄어든다.

2. JSTL 코어 태그

❖ 코어 태그 라이브러리

- 변수 설정이나 if-else 와 같은 논리 처리에 사용되는 스크립트 코드를 대체하는 태그를 제공한다.

❖ taglib 디렉티브 추가

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

기능 분류	태그	설 명
변수 지원	set	JSP에서 사용할 변수를 설정한다
	remove	설정한 변수를 제거한다
흐름 제어	if	조건에 따라 내부 코드를 수행한다
	choose	다중 조건을 처리할 때 사용된다
	forEach	컬렉션이나 Map의 각 항목을 처리할 때 사용된다
	forTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용된다
URL 처리	import	URL을 사용하여 다른 자원의 결과를 삽입한다
	redirect	지정한 경로로 리다이렉트 한다
	url	URL을 재작성 한다
기타 태그	catch	익셉션을 처리할 때 사용한다
	out	JspWriter에 내용을 출력한다

2. JSTL 코어 태그

❖ 변수 지원 태그 - <c:set> 태그

- <c:set> 태그는 EL 변수의 값이나 EL 변수의 프로퍼티 값을 지정할 때 사용

❖ <c:set> 태그 사용 형식 ①

```
< c:set var="변수명" value="값" [scope="영역"] /> 또는  
< c:set var="변수명" [scope="영역"] >값 </c:set>
```

- **var** : 값을 지정할 EL 변수의 이름을 지정
- **value** : 변수의 값을 지정. 표현식, EL, 정적인 텍스트 사용 가능
- **scope** : 변수를 저장할 영역을 지정. 기본값은 page

2. JSTL 코어 태그

❖ 변수 지원 태그 - <c:set> 태그

- <c:set> 태그는 EL 변수의 값이나 EL 변수의 프로퍼티 값을 지정할 때 사용

❖ <c:set> 태그 사용 형식 ②

- 객체의 프로퍼티 값을 설정

```
< c:set target="대상" property="프로퍼티이름" value="값" />  
또는  
< c:set target="대상" property="프로퍼티이름" >값 </c:set>
```

- **target** : 프로퍼티 값을 설정할 대상 객체를 지정. 표현식, EL 변수 사용 가능.
 대상 객체는 자바빈 객체나 Map이어야 함
- **property** : 설정할 프로퍼티의 이름을 지정. target이 자바빈 객체인 경우 프로퍼티 이름에 해당하는 set 메소드를 제공
- **value** : 프로퍼티의 값을 지정

2. JSTL 코어 태그

❖ 변수 지원 태그 - <c:remove> 태그

- <c:remove> 태그는 <c:set> 태그로 지정한 변수를 삭제할 때 사용

❖ <c:remove> 태그 사용 형식

```
< c:remove var="varName" [scope="영역"] />
```

❖ <c:remove> 태그 사용시 주의 사항

- 삭제할 변수의 scope을 지정하지 않으면 동일한 이름으로 저장된 모든 영역 (request, response, page, application)의 변수를 삭제한다

```
< c:set var="name" value="홍길동" scope="request" />
< c:set var="name" value="홍길동" scope="session" />
< c:remove var="name" />
```

scope을 지정하지 않고 <c:remove> 태그를 사용하면 모든 영역에 저장된 동일한 이름의 변수를 모두 삭제한다

2. JSTL 코어 태그

❖ 흐름 제어 태그

- JSTL에서 많이 사용되는 태그 중 하나
- if, choose, forEach, forTokens 네 가지 태그 제공

❖ <c:if> 태그

- 자바 언어의 if 블록과 유사한 기능을 제공

```
< c:if test="조건" >
    . . . .
< /c:if>
```

2. JSTL 코어 태그

❖ 코어 태그 라이브러리 예제 (jstlCore.jsp)

```
jstlCore.jsp jstlResult.jsp jstlCore2.jsp jstlResult2.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<!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6<body>
7     <c:set var="name"    value="홍길동" scope="request"></c:set>
8     <c:set var="gender" value="남자"   scope="request"></c:set>
9     <c:set var="age"    value="25"    scope="request"></c:set>
10    <c:set var="array"  value="<%=new int[] {1,2,3,4,5} %>" scope="request"/>
11    <jsp:forward page="jstlResult.jsp"></jsp:forward>
12 </body></html>
13
```

2. JSTL 코어 태그

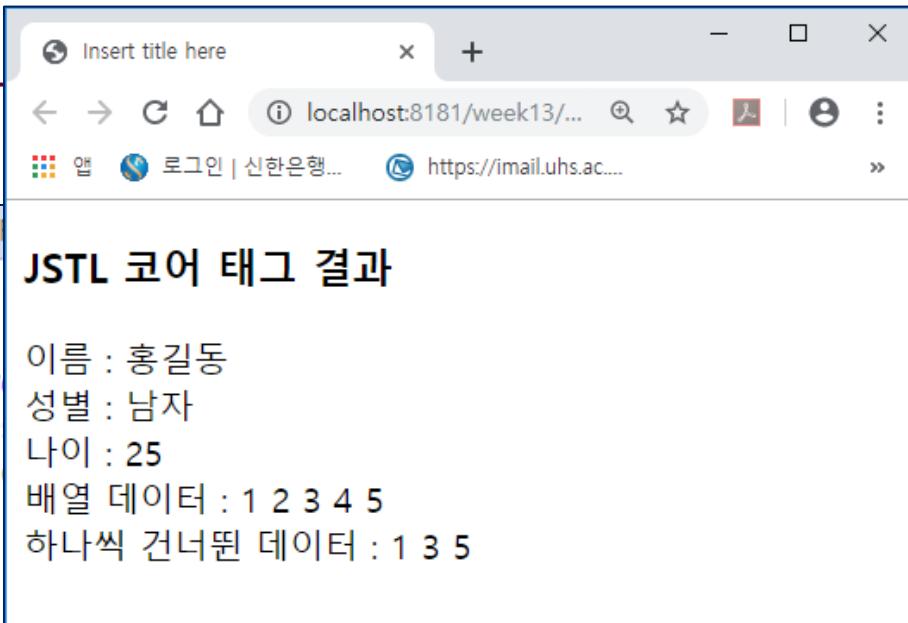
❖ 코어 태그 라이브러리 예제 (jstlResult.jsp)

```
jstlCore.jsp jstlResult.jsp jstlCore2.jsp jstlResult2.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<!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6<body>
7     <h3>JSTL 코어 태그 결과</h3>
8     이름 : ${name } <br>
9     성별 : ${gender }<br>
10    나이 : ${age } <br>
11    배열 데이터 :<c:forEach var="data" items="${array}">
12        ${data }
13        </c:forEach><br>
14    하나씩 건너뛴 데이터 :
15    <c:forEach var="data" begin="0" end="4" step="2" items="${array}">
16        ${data }
17        </c:forEach><br>
18 </body></html>
```

2. JSTL 코어 태그

❖ 코어 태그 라이브러리 예제 (jstlR

```
jstlCore.jsp jstlResult.jsp jstlCore2.jsp jstl
1 <%@ page language="java" contentType=
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://jav
4<!DOCTYPE html><html><head><meta char
5 <title>Insert title here</title></hea
6<body>
7     <h3>JSTL 코어 태그 결과</h3>
8     이름 : ${name } <br>
9     성별 : ${gender }<br>
10    나이 : ${age } <br>
11    배열 데이터 :<c:forEach var="data" items="${array}">
12        ${data }
13        </c:forEach><br>
14    하나씩 건너뛴 데이터 :
15    <c:forEach var="data" begin="0" end="4" step="2" items="${array}">
16        ${data }
17        </c:forEach><br>
18 </body></html>
```



2. JSTL 코어 태그

❖ 코어 태그 라이브러리 예제 (jstlCore2.jsp)

```
jstlCore.jsp jstlResult.jsp jstlCore2.jsp jstlResult2.jsp
1 <%@page import="week13.MemberInfo"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
5<!DOCTYPE html><html><head><meta charset="UTF-8">
6 <title>Insert title here</title></head>
7<body>
8<%
9   MemberInfo member = new MemberInfo();
10<%
11   <c:set var="m" value="<%=>member %>" scope="request"/>
12   <c:set target="${m }" property="id" value="admin"/>
13   <c:set target="${m }" property="password" value="admin"/>
14   <c:set target="${m }" property="name" value="김철수"/>
15   <jsp:forward page="jstlResult2.jsp"/>
16 </body></html>
```

2. JSTL 코어 태그

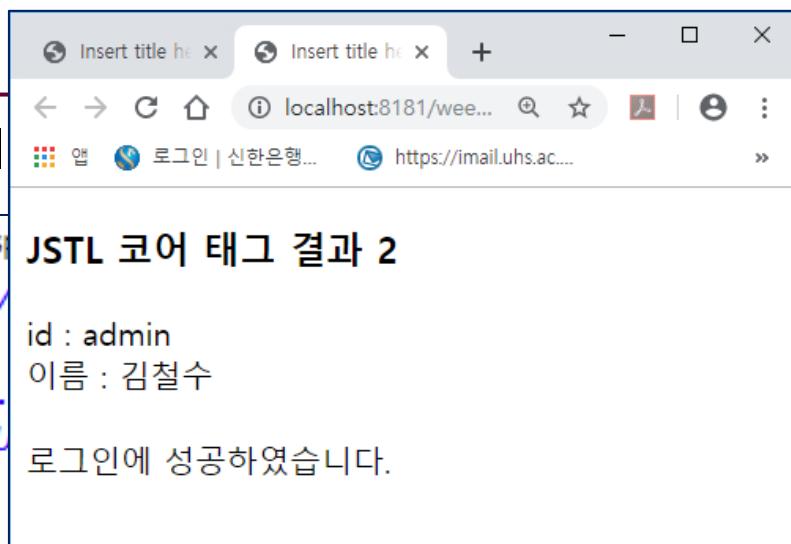
❖ 코어 태그 라이브러리 예제 (jstlResult2.jsp)

```
jstlCore.jsp jstlResult.jsp jstlCore2.jsp jstlResult2.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<!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6<body>
7     <h3>JSTL 코어 태그 결과 2</h3>
8     id : ${m.getId() } <br>
9     이름 : ${m.getName() } <br><br>
10
11<c:if test="${m.getId() == m.getPassword() }">
12    로그인에 성공하였습니다.
13</c:if>
14<c:if test="${m.getId() != m.getPassword() }">
15    로그인에 실패하였습니다.
16</c:if>
17</body></html>
```

2. JSTL 코어 태그

❖ 코어 태그 라이브러리 예제 (jstlResult.jsp)

```
1  <%@ page language="java" contentType="text/
2      pageEncoding="UTF-8"%>
3  <%@ taglib prefix="c" uri="http://java.sun.
4  <!DOCTYPE html><html><head><meta charset="U
5  <title>Insert title here</title></head>
6  <body>
7      <h3>JSTL 코어 태그 결과 2</h3>
8      id : ${m.getId() } <br>
9      이름 : ${m.getName() } <br><br>
10
11     <c:if test="${m.getId() == m.getPassword() }">
12         로그인에 성공하였습니다.
13     </c:if>
14     <c:if test="${m.getId() != m.getPassword() }">
15         로그인에 실패하였습니다.
16     </c:if>
17 </body></html>
```



```
<h3>JSTL 코어 태그 결과 2</h3>
id : ${m.id }<br>
이름 : ${m.name }<br>

<c:if test="${m.id == m.password }">
    로그인에 성공했습니다.
</c:if>
<c:if test="${m.id != m.password }">
    로그인에 실패했습니다.
</c:if>
```

2. JSTL 코어 태그

- ❖ <c:choose>, <c:when>, <c:otherwise> 태그
 - <c:choose>태그는 자바의 switch 구문과 if-else 블록을 혼합한 형태
 - 다수의 조건문을 하나의 블록에서 수행할 때 사용

```
< c:choose>
    <c:when test="${member.level == 'trial'}">
        ...
    </c:when>
    <c:when test="${member.level == 'regular'}">
        ...
    </c:when>
    <c:when test="${member.level == 'provider'}">
        ...
    </c:when>
    <c:otherwise>
        ...
    </c:otherwise>
< /c:choose>
```

- test 조건식이 true인 경우에 <c:when> 태그의 몸체 내용을 실행
- 한 개의 조건을 만족하면 나머지 태그는 실행되지 않음
- 모든 조건을 만족하지 않으면 otherwise 태그를 실행

2. JSTL 코어 태그

❖ <c:choose>, <c:when>, <c:otherwise> 태그

- chooseTag.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>Insert title here</title></head>
<body>
    <h3>JSTL 코어 choose 태그 결과</h3>
    <ul>
        <c:choose>
            <c:when test="${param.name == '홍길동' }">
                <li>당신의 이름은 ${param.name }입니다
            </c:when>
            <c:when test="${param.age > 20 }">
                <li>당신은 20세 이상입니다
            </c:when>
            <c:otherwise>
                <li>당신은 홍길동이 아니고 20세 이하입니다
            </c:otherwise>
        </c:choose>
    </ul>
</body>
```

2. JSTL 코어 태그

❖ <c:forEach> 태그

- <c:forEach> 태그는 배열, Collection 또는 Map에 저장되어 있는 값들을 순차적으로 처리할 때 사용
- 자바의 for, do-while 등을 대신해서 사용할 수 있다.

❖ <c:forEach> 태그의 사용 형식

```
< c:forEach var="변수" items="아이템" >
```

```
...
```

```
 ${변수 사용}
```

```
...
```

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

- items : Map, 배열, Collection이 올 수 있다.
- items 가 배열인 경우에는 객체의 배열 뿐만 아니라 기본 데이터 타입의 배열도 처리 가능 - wrapper class(Integer, Double 등) 사용

2. JSTL 코어 태그 <c:forEach> 태그 (forEach.jsp)

```
<%@page import="java.util.Date"%>
<%@page import="java.util.HashMap"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<body>
<%
    HashMap<String, Object> mapData = new HashMap<String, Object>();
    mapData.put("id", "admin");
    mapData.put("today", new Date());
%>
<c:set var="map" value="<%=mapData %>" />
<c:forEach var="m" items="${map }">
    ${m.key } = ${m.value }<br>
</c:forEach>
```

```
<hr>
<h3>구구단 : 7단</h3>
<ul>
    <c:forEach var="num" begin="1" end="9">
        <li> 7 * ${num } = ${7*num }
    </c:forEach>
</ul>
</body>
```

2. JSTL 코어 태그

❖ <c:forTokens> 태그

- <c:forTokens>태그는 `java.util.StringTokenizer` 클래스와 같은 기능 제공

❖ <c:forTokens> 태그의 사용 형식

```
< c:forTokens var="token" items="문자열" delims="구분자">
    ${token}의 사용
< /c:forTokens>
```

- **items** : **items** 속성으로 전달 받은 문자열을 구분자를 이용해서 나눈다.
- 구분한 각 문자열을 **var** 속성에 명시한 변수에 저장한다.

2. JSTL 코어 태그

❖ <c:forTokens> 태그 (tokenTag.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>Insert title here</title></head>
<body>
    <h3>forTokens 태그 </h3>
    콤마와 점을 구분자로 사용 : <br>
    <c:forTokens var="token"
        items="빨강, 주황. 노랑. 초록, 파랑, 남색, 보라"
        delims=",. ">
        ${token }
    </c:forTokens>

</body>
</html>
```

3. JSTL formatting 태그

❖ **formatting 태그 라이브러리**

- 특정 지역에 따라 알맞은 메시지를 출력해야 할 때 사용
- JSP 페이지에서 다양한 언어에 맞는 메시지 출력 기능을 제공
- 한글 브라우저에서 접속 – 한글 메시지 출력
- 영문 브라우저에서 접속 – 영문 메시지 출력

3. JSTL formatting 태그

❖ taglib 디렉티브 추가

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

기능 분류	태그	설명
로케일 지정	setLocale	Locale을 지정한다
	requestEncoding	요청 파라미터의 캐릭터 인코딩을 지정한다
메시지 처리	bundle	사용한 번들을 지정한다
	message	지역에 알맞은 메시지를 출력한다
	setBundle	리소스 번들을 읽어와 특정 변수에 저장한다
숫자 및 날짜 포맷팅	formatNumber	숫자를 포맷팅한다
	formatDate	Date 객체를 포맷팅한다
	parseDate	문자열로 표시된 날짜를 분석해서 Date 객체로 변환한다
	parseNumber	문자열로 표시된 날짜를 분석해서 숫자로 변환한다
	setTimeZone	시간대 정보를 특정 변수에 저장한다
	timeZone	시간대를 지정한다

3. JSTL formatting 태그

❖ 로케일 지정 태그

- <fmt:setLocale> : 국제화 태그들이 사용할 로케일을 지정
- <fmt:requestEncoding> : 요청 파라미터의 캐릭터 인코딩을 지정

- 웹 브라우저는 Accept-Language 헤더를 사용해서 수용 가능한 언어 목록을 전송한다.
- JSTL 국제화 태그들은 이 헤더의 값을 사용해서 언어별로 알맞은 처리를 실행한다.

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

- **value** : Locale을 “언어코드_국가코드” 형식으로 지정

3. JSTL formatting 태그

❖ 숫자 및 날짜 포맷팅 처리 태그

- <fmt:formatNumber> : 숫자 출력
- <fmt:parseNumber> : 숫자 파싱
- <fmt:formatDate> : 날짜 출력
- <fmt:parseDate> : 날짜 파싱
- <fmt:setTimeZone>, <fmt:timeZone> : 시간대 설정

3. JSTL formatting 태그

❖ 국제화 태그 라이브러리 예제 (jstlFormat.jsp)

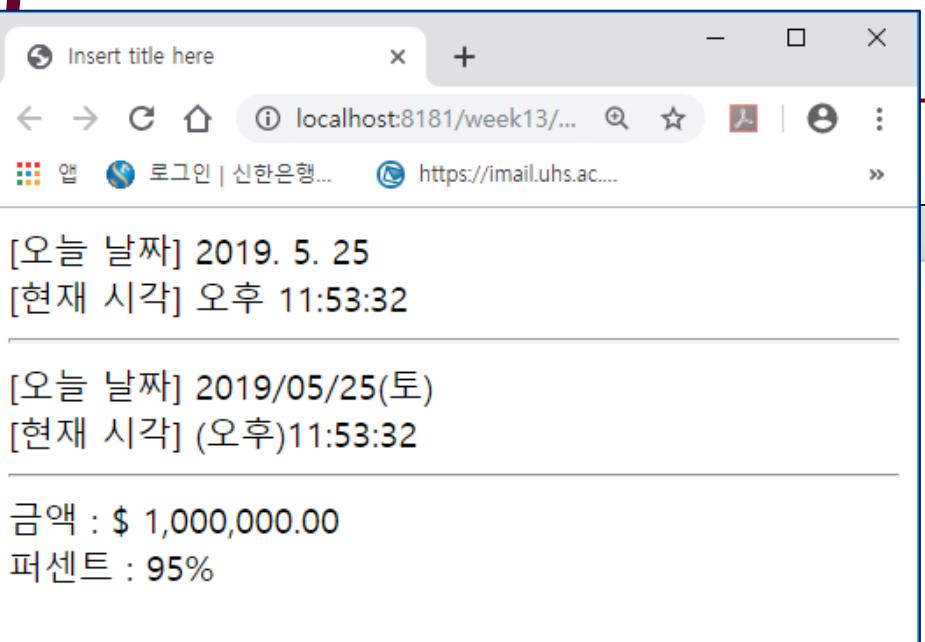
```
jstlFormat.jsp ✎
1 <%@page import="java.util.Date"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3     pageEncoding="UTF-8"%>
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
5 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
6<!DOCTYPE html><html><head><meta charset="UTF-8">
7 <title>Insert title here</title></head>
8<body>
9     <c:set var="date" value="<%=new Date()%>" />
10    [오늘 날짜] <fmt:formatDate value="${date }"/><br>
11    [현재 시각] <fmt:formatDate value="${date }" type="time"/><hr>
12    [오늘 날짜] <fmt:formatDate value="${date }" type="date"
13                                pattern="yyyy/MM/dd(E)"/><br>
14    [현재 시각] <fmt:formatDate value="${date }" type="time"
15                                pattern="(a)hh:mm:ss"/><hr>
16    금액 : <fmt:formatNumber value="1000000" type="currency"
17                                currencySymbol="$"/><br>
18    퍼센트 : <fmt:formatNumber value="0.95" type="percent"/>
19 </body></html>
```

3. JSTL formatting 태그

❖ 국제화 태그 라이브러리 예제

jstlFormat.jsp

```
1 <%@page import="java.util.Date"%>
2 <%@ page language="java" content-
   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http:-
4 <%@ taglib prefix="fmt" uri="htt-
6<!DOCTYPE html><html><head><meta
7 <title>Insert title here</title>
8<body>
9   <c:set var="date" value="<%=new Date()%"/>
10  [오늘 날짜] <fmt:formatDate value="${date }"/><br>
11  [현재 시각] <fmt:formatDate value="${date }" type="time"/><hr>
12  [오늘 날짜] <fmt:formatDate value="${date }" type="date"
13                           pattern="yyyy/MM/dd(E)"/><br>
14  [현재 시각] <fmt:formatDate value="${date }" type="time"
15                           pattern="(a)hh:mm:ss"/><hr>
16  금액 : <fmt:formatNumber value="1000000" type="currency"
17                           currencySymbol="$"/><br>
18  퍼센트 : <fmt:formatNumber value="0.95" type="percent"/>
19 </body></html>
```



3. JSTL formatting 태그

❖ 국제화 태그 라이브러리 예제 (jstlFormat.jsp)

```
8<body>
9    <c:set var="date" value="<%=new Date()%>" />
10   [오늘 날짜] <fmt:formatDate value="${date }"/><br>
11   [현재 시각] <fmt:formatDate value="${date }" type="time"/><hr>
12   [오늘 날짜] <fmt:formatDate value="${date }" type="date"
13           pattern="yyyy/MM/dd(E)"/><br>
14   [현재 시각] <fmt:formatDate value="${date }" type="time"
15           pattern="(a)hh:mm:ss"/><hr>
16   금액 : <fmt:formatNumber value="1000000" type="currency"
17           currencySymbol="$"/><br>
18   퍼센트 : <fmt:formatNumber value="0.95" type="percent"/><hr>
19   <fmt:setLocale value="en"/>
20   지역 설정 변경 : 미국<br>
21   [오늘 날짜] <fmt:formatDate value="${date }"/><br>
22   [현재 시각] <fmt:formatDate value="${date }" type="time"/><hr>
23   <fmt:setLocale value="ja_jp"/>
24   지역 설정 변경 : 일본<br>
25   [오늘 날짜] <fmt:formatDate value="${date }"/><br>
26   [현재 시각] <fmt:formatDate value="${date }" type="time"/>
27 </body></html>
```

3. JSTL formatting 태그

❖ 국제화 태그 라이브러리 예제 (jstlForm)

```
8<body>
9    <c:set var="date" value="<%=new Date()%>" %>
10   [오늘 날짜] <fmt:formatDate value="${date}" %>
11   [현재 시각] <fmt:formatDate value="${date}" %>
12   [오늘 날짜] <fmt:formatDate value="${date}" %>
13       pattern="yyyy/MM/dd"
14   [현재 시각] <fmt:formatDate value="${date}" %>
15       pattern="(a)hh:mm:ss"
16   금액 : <fmt:formatNumber value="10000000" %>
17       currencySymbol=
18   퍼센트 : <fmt:formatNumber value="0.95" type="percent" %>
19   <fmt:setLocale value="en"/>
20   지역 설정 변경 : 미국<br>
21   [오늘 날짜] <fmt:formatDate value="${date}" %>
22   [현재 시각] <fmt:formatDate value="${date}" %>
23   <fmt:setLocale value="ja_jp"/>
24   지역 설정 변경 : 일본<br>
25   [오늘 날짜] <fmt:formatDate value="${date}"/><br>
26   [현재 시각] <fmt:formatDate value="${date}" type="time"/>
27 </body></html>
```

[오늘 날짜] 2019. 5. 26
[현재 시각] 오전 12:02:02

[오늘 날짜] 2019/05/26(일)
[현재 시각] (오전)12:02:02

금액 : \$ 1,000,000.00
퍼센트 : 95%

지역 설정 변경 : 미국
[오늘 날짜] May 26, 2019
[현재 시각] 12:02:02 AM

지역 설정 변경 : 일본
[오늘 날짜] 2019/05/26
[현재 시각] 0:02:02

최신 JSP 트렌드

❖ JSP 역할

항목	과거 방식	현재 권장 방식
코드 작성	스크립틀릿(<code><% %></code>)	JSTL + EL 조합
데이터 접근	<code>request.getAttribute()</code>	<code>\${requestScope.xxx}</code>
제어문	if/for 자바 코드	<code><c:if></code> , <code><c:forEach></code>
출력	<code>out.println()</code>	<code><c:out></code> 또는 <code>\${}</code>
MVC 구성	JSP + 서블릿 혼합	JSP는 View 전용, Controller가 데이터 전달

스프링 부트 환경 구축

- ❖ 스프링 전용 STS(Spring Tool Suite) 다운로드
 - <https://spring.io/tools>

Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free.
Open source.

4.23.1 - LINUX X86_64

4.23.1 - LINUX ARM_64

4.23.1 - MACOS X86_64

4.23.1 - MACOS ARM_64

4.23.1 - WINDOWS X86_64

