

14. 액션태그 (Action Tag) 2

1) 태그 사용을 위한 라이브러리

(1) **JSTL** : Jsp Standard Tag Library

- > 액션 태그를 사용할 수 있도록 해주는 라이브러리 형태로 제공된 자바 패키지
- > <http://jakarta.apache.org> → tab lib 를 받는 곳
- > jstl.jar / standard.jar 두 개를 웹상에서 다운로드 받아
 - 전역 : C:\W\Tomcat 8.5\lib 위치에 복사해두면 모든 프로젝트에서 불러다 사용
 - 특정 프로젝트에서만 사용할 경우 : WEB-INF → lib 에 복사

(2) **core**

- > 변수선언, 객체의 값 불러오기, 제어문 등 가장 많이 쓰이는 액션태그

(3) **fmt**

- > 날짜, 시간, 통화 등에 쓰이는 액션 태그

(4) 기타로 sql, xml , fn 드이 있음.

2) el(Expression Language)

: 변수의 값을 출력, 객체의 멤버변수의 값 출력하는 태그

표현언어 → 출력문(변수의 값, 수식계산, 객체의속성값 출력)

형식)

`${표현식} → out.println(변수명); : out.println(str);`
`<%=str%> → ${str}`

ex) `<%=article.getNum()%>` → `${article.num}` → **`${객체명.멤버변수명}`**

`${article['num']}`

`${article["num"]}`

ex2) `out.println(4+5);` `<%=4+5%>` `${4+5}` → 모두 같은 기능

3) 태그 설정

형식)

`<%@taglib prefix="식별자(액션태그의 종류)" uri="액션태그의 선언문(core)" %>`

- > 위와 같은 선언문 형식으로 '식별자'와 그 태그의 라이브러리 위치 'uri'를 선언해준다.

jstlcore.jsp : core tag 의 사용법

```
<%@ 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>jstlcore.jsp(변수(객체) 사용 범위, out태그)</title>
</head>
<body>
<h1>출력문</h1>
<h3>
<%
    out.println(1+2);
%>
<br>
<%=1+2%> <!-- 표현식을 이용한 출력 -->
<hr>
"${1+2}" : ${1+2}<br><!-- 표현(el)언어를 이용한 출력 -->
<c:out value="${1+2}"/> <!-- 액션태그와 표현(el)언어를 이용한 출력 -->
${1>3} <c:out value="${1>3}"/>
${1 gt 3} <c:out value="${1 gt 3}"/> <!-- html 에서 부등호 기호를 사용하기 -->

<h2>사용범위</h2>
<!--el 표현언어 사용 시에는 같은 변수의 값을 출력하는 경우 (저장영역을 표시.출력할 변수명) -->
<-- ↓ core 태그 사용시 사용범위 -->
<c:set var="name" value="hello" scope="page"/><br>
c:out name:<c:out value="${name}"/><br>
<c:set var="name" value="하늘" scope="session"/>
expression name: <%=session.getAttribute("name") %><br>
session 객체의 값 name:<c:out value="${sessionScope.name}"/>

remove session scope var "name" : <c:remove var="name" scope="session"/> <br>
session 객체의 값 name2:<c:out value="${sessionScope.name}"/>
</h3>
</body>
</html>
```

`sessionScope.name`

el 내장 객체의 사용범위
(jsp 내의 범위와 동일)

- pageScope (default)
- requestScope
- sessionScope
- applicationScope

사용범위를 다르게 주어 위의 name
과는 다른 개체가 된다.

```
<%@ 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>JSTL사용(액션태그2) core (set태그, remove 태그)</title>
```

```
</head>
```

```
<body>
```

```
<%
```

```
    //int num1 = 20;
```

```
    //double num2 = 10.5;
```

```
    //java.util.Date today = new java.util.Date();
```

```
%>
```

<!-- <jsp:useBean ~~~ : jsp는 식별자 / 다른 태그를 사용할 때는 식별자 또한 그에 맞는 식별자를 사용한다 -->

```
<c:set var="num1" value="${20}" scope="page"/>
```

```
<c:set var="num2">
```

```
    10.5
```

```
</c:set>
```

```
<c:set var="today" value="<%=new java.util.Date()%>"/>
```

```
<hr>
```

```
변수 num1=${num1} <br>
```

```
변수 num2=${num2} <br>
```

```
num1+num2=${num1+num2}<br>
```

```
오늘은 ${today} 입니다.<p>
```

```
<c:remove var="num1" scope="page" />
```

```
삭제한 후의 num1 변수 num1 = ${num1}<br>
```

```
num1+num2=${num1+num2}
```

```
</body>
```

```
</html>
```

형식) 변수선언

set 액션 태그

var = "변수명" value="저장할 값" (자료형은 자동부여) scope="사용범위"

el 문법으로 꺼내 쓰기

형식) 변수 삭제

remove 액션 태그

var = "변수명" scope="사용범위"

setTag2.jsp (c:set 태그 / c:param 태그)

```
<%@ 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="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<fmt:requestEncoding value="utf-8"/>
<c:set var="map" value="<%=new java.util.HashMap() %>" />
<html>
<head>
<meta charset="UTF-8">
<title>setTag2의 2번째 예제</title>
</head>
<body>

<c:set target="${map}" property="name" value="박종혁"/>
변수 map에 저장된 name값: ${map.name} <br>
변수 map에 저장된 name값: ${map['name']} <br>
변수 map에 저장된 name값: ${map["name"]} <br>

<form method="post" action="setTag2.jsp" <!-- action="test.jsp" -->
    이름:<input type="text" name="name">
    <input type="submit" value="전송">
</form>
<hr>
이름은 <%=request.getParameter("name") %>입니다.
이름은(el) ${param.name} 입니다.
</body>
</html>
```

날짜, 시간, 통화, 언어

<c:set target = 객체명 property="속성이름" value="값"/>

- 만들어져 있는 객체에 값을 넣을 때 사용
- 외부에서 만들어진 객체를 쓰는 경우엔 \${객체명}으로 인식
- <jsp:setProperty name="객체명" value="값"/>

모두 같은 의미이지만 \${map.name}이
가장 간단한 표현이다

자기자신에게

<c:param name="전달매개변수명" value="값">

- el언어에 내장된 객체
- 매개변수를 가져올 때 관여하는 내장 객체
- jsp 의 내장객체인 request를 대신한다.
- : property 가 name인 파라미터를 가져와 표시

ifTag (if 태그 사용)

```
<%@ 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>if문 태그</title>
</head>
<body>
<%
    //int age=20;
    //String name="bk";
    //if(조건식) 실행문 if(age>19) => 형식: c:if test="조건식" or "${조건식}"
    // 즉 c(core) 태그의 속성 if를 사용하는데 조건식test를 쓸거야
%>
<c:set var="age" value="${20}"/>
<c:set var="name" value="'bk'"/> <!-- 문자를 표현언어로 넣을 때 -->
<c:if test="true">
    무조건 실행돼서 화면에 결과를 보여준다.<br>
</c:if>
<c:if test="${19<age}">
    당신의 나이는 19세 이상이군요.<br>
</c:if>

<c:if test="${name=='bk'}">
    name 파라미터 값은 ${name}입니다.<br>
</c:if>
<!-- if 태그는 else 구문을 처리해주는 속성이 없다. 그 반대의 조건을 따로 주는 c:if 문을 따로 적어주어야한다
-->
<c:if test="${name!='bk'}">
    name 파라미터 값은 bk가 아닌 ${name}입니다.<br>
</c:if>
</body>
</html>
```

<c:if test="조건식" or "\${조건식}"></c:if>

→ 기존 if문과 같은 기능이다

→ 하지만 **else** 를 쓸 수 없다.

→ **else**는 또 하나의 **<c:if test~>** 를 추가해서 다른 조건으로 처리함

param.jsp (c:param 태그 / redirect 태그)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<c:redirect url="chooseTag.jsp">
    <c:param name="age2" value="20" />
</c:redirect>
```

response.sendRedirect("주소"); > 외부사이트로 이동(데이터공유 없이)
ex) response.sendRedirect("choesTag.jsp?name2=bk");

→ 태그로 표현 :

형식 1) <c:redirect url ="이동할 페이지"/> (매개변수 없음)

형식 2) <c:redirect url ="이동할 페이지"> (매개변수 있음)

<c:param name="전달할 매개변수명" value="값"/>

</c:redirect>

catch.jsp (c:catch 태그 → 예외처리 구문)

```
<%@ 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>예외를 처리해주는 액션태그</title>
</head>
<body>
<c:catch var="ex">
    name의 매개변수명 : <%=request.getParameter("name") %> <br>
    <%
        if(request.getParameter("name").equals("test")){%>
            ${param.name}은 test 입니다.
        <% } %>
    </c:catch>
    <!-- <c:if test="${ex!=null}"> ex==null (empty ex) --%>
    <c:if test = "${not empty ex}"> <!-- ex에 정보가 들어있다면, -->
        예외가 발생하였습니다. <br> ${ex}
    </c:if>
</body>
</html>
```

java 에서의 예외처리를 태그로 표현

형식)

<c:catch var="예외처리 객체명">

예외 발생시 처리할 내용

</c:catch>

name의 매개변수명 : null
예외가 발생하였습니다.
java.lang.NullPointerException

forEach.jsp (c:forEach 태그 → 반복문)

```
<%@ 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>
<%
```

```
    java.util.HashMap mapData = new java.util.HashMap();
    mapData.put("name", "연습");
    mapData.put("today", new java.util.Date()); // 날짜 객체 저장
%>
```

반복문으로 배열에 있는 값을 출력하기 위해
HashMap 객체를 선언
"name" → "연습" 값을 넣는다.
"today" → "날짜객체"
int형 배열 intArray → 1,2,3,4,5 값을 넣어둔다.

```
    <c:set var="intArray" value="<%=new int[] {1,2,3,4,5} %>"/>
    <c:set var="map" value="<%=mapData %>"/> <!-- map이 mapData와 같은 주소를 갖게 됨 -->
```

```
<html>
<head>
```

```
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
```

태그로 반복문을 사용

```
<c:forEach var="첨자변수" begin="초기값" end="마지막값" step="증가값">
    반복문 내에서의 작업
</c:forEach>
```

```
<h4> 1~100 까지의 홀수의 합</h4>
```

```
<c:set var="sum" value="0"/> <!-- <c:set var="sum" value="{0}"/> --%>
```

```
<c:forEach var="i" begin="1" end="100" step="2">
```

```
    <var c:if test="{i%2!=0}">
```

```
        <c:set var="sum" value="{sum+i}"/>
```

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

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

```
결과 : {sum}
```

```
<hr>
```

```
<!--
```

```
<c:set var="intArray" value="<%=new int[] {1,2,3,4,5} %>"/>
--%>
```

```
<h4> 컬렉션객체 및 배열의 값을 출력(확장 for)</h4>
```

```
<c:forEach var="i" items="{intArray}" begin="1" end="3">
```

```
    [{i}]
```

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

```
<h4> Map객체를 이용(key,value)</h4>
```

```
<!-- Map객체의 i값은 Map에 저장된 각각의 객체를 의미 -->
```

```
<c:forEach var="i" items="{map}">
```

```
    {i.key}={i.value} <!-- 키이름 = 키값 -->
```

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

```
</body>
```

```
</html>
```

[0] = 1 , [1] = 2,,, [4]=5
중에 1~3 인덱스 값을 출력

출력할 배열개체