

## 6. JSTL

### 1.JSTL

JSP 는 개발자가 직접 <jsp:include>와 같은 태그를 작성할 수 있는 기능을 제공하는데 이를 커스텀태그라고한다. 커스텀태그중에서 우리가 많이사용하는것들을 모아서

JSTL(JSP Standard Tag Library)라는 규약을 만들었다

JSTL을 사용하면 보다 깔끔하고 이해하기쉬운 JSP 코드를 작성할수있다.

톰캣 8.0 은 기본적으로 지원하지는 않으며 JSTL API 를 제공하는 jar 파일을 별도로 구해야 한다.

-JSTL 셋팅하기

\* 설치

다운로드 사이트:

<http://tomcat.apache.org/taglibs/standard/>

1. 위 사이트에서 JSTL 을 다운 받습니다.
2. 를 복사합니다.
3. Tomcat 의 디렉토리 내의 `context\WEB_INF\lib` 안에 붙여넣기.

\* JSP 선언

JSP 에 taglib 추가법

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

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

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

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

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

### 2.JSTL이 제공하는 태그의 종류

## 2-1.코어 태그

코어 태그라이브리는 if-else 나 변수의지정 등과같이 논리적인 처리를 위해서 사용되는 스트림트코드를 대체하기위한 태그를 제공하는데 제공하는 태그는 다음과같다.

기능 분류	태그	설명
변수 지원	set	jsp에서 사용될 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.
	forEachTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용된다.
URL처리	import	URL을 사용하여 다른 자원의 결과를 삽입한다.
	redirect	지정한 경로로 리다이렉트한다.
	url	URL을 재작성한다.
기타 태그	catch	예외 처리에 사용된다.
	out	JspWrite에 내용을 알맞게 처리한 후 출력한다.

### -변수지원태그

변수지원태그는 JSTL 이지원하는 태그에서 사용할 수 있는 변수를 설정하기위해 사용된다.

변수지원태그에는 set 태그와 remove 태그가있는데 set 태그부터살펴보자.

set 태그는 EL변수의 값이나 EL 변수의 프로퍼티값을 지정할 때 사용된다.

page,request,session,application의 네영역에 저장되어있는 객체를 <jsp:useBean>

태그를 와 <jsp:setProperty>태그를 사용하여 JSP 코드내에 사용될변수를 지정하고

변수 프로퍼티의 값을 지정할 수 있는 것과 마찬가지로,set 태그는 JSTL태그에서 사용될 변수의 값을 처리하는데 사용된다.

### - <c:set > tag

사용형태

```
<c:set var="varName" scope="session" value="someValue"/>
```

각속성의 값

- var- 값을 지정할 변수의 이름
- scope-변수를 저장할 영역을 지정한다.값은 page,request,session,application 중의 하나가 온다.
- Value-변수의 값을 지정한다. 표현식이나 EL 을 사용하여 값을 지정할수있다.

Value 속성을 지정하지않고 다음과같이 태그의 몸체에 값을 입력할수도있다.

```
<c:set var="varName" scope="request">
someValue
</c:set>
```

#### - <c:remove> tag

set 태그로 지정된변수를 삭제할때사용

형식:

```
<c:remove var="varName" scope="request"/>
```

- var- 값을 지정할 변수의 이름(삭제할변수이름)
- scope-변수를 저장할 영역을 지정한다.값은 page,request,session,application 중의 하나가 온다.(삭제변수가 저장된영역)

주의

set tag 에서 지정한영역과 remove tag 에서 지정한영역이 다를경우 변수는 제거안됨

```
<c:set var="member" scope="session" value="guard"/>
```

.

```
<c:remove var=" member" scope="request"/>
```

setTag.jsp

```
<%@ page language="java" contentType="text/html;
charset=EUC-KR"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="num1" value="${20}"/>
<c:set var="num1">
10.5
</c:set>
<c:set var="today" scope="session" value="<%= new
java.util.Date() %>"/>
<html>
<head>
<title>set태그와 remove 태그</title>
</head>
<body>
<hr>
set
<hr>
변수 num1:${num1}<br>
변수 num2:${num2}<br>

num1+num2=${num1+num2}<br>
오늘은 ${pageScope.today }<br>
<hr>
remove
<hr>
<c:remove var="num1" scope="page"/>
삭제한후의 num1=${num1}<br>
삭제한후의 num1+num2=${num1+num2}<br>

</body>
</html>
```

set 태그를 사용하면 target 속성과 property 속성을 통해서 자바빈 프로퍼티의 값이나 Map 의 키값을 지정할수있다.

```
<c:set var="member" scope="request" value="<%= new Member()%>" />
..
.
<c:set target ="${member}" property="name" value="김" />
```

#### -흐름제어 태그

JSTL에서 가장많이 사용되는 태그중의 하나는 흐름제어태그이다.

JSP 코드에서 스크립트코드를 사용할경우 if-else 블록이나 for 블록등이 조금만 중첩되더라도 <% %>,{,}등이 복잡하게 얽히게 되며 이는 JSP 코드를 읽는 사람을 괴롭히는 요소가 된다.이런 코드의 복잡함을 없애기 위해서 JSTL 태그를 사용하여 흐름제어를 사용할수 있도록 하였다.

#### - <c:if> tag

형식:

```
<c:if test="조건"/>
.
</c:if>
```

test 속성의 값에는 true 나 false 를 리턴하는 조건문이 오는데 이조건문의 결과값이 true 이면 if 블록이 처리된다.

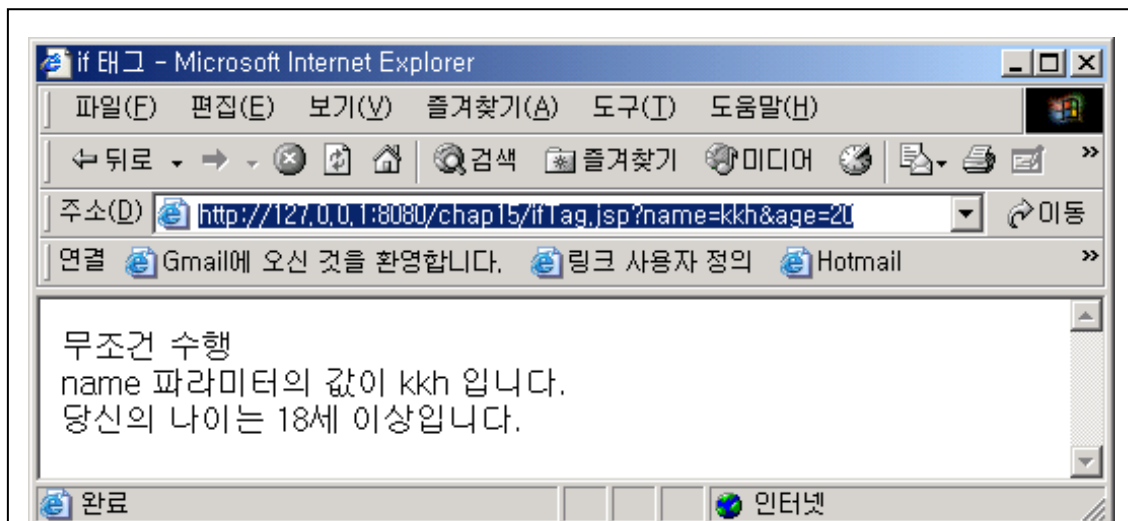
ifTest.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head><title>if 태그</title></head>
<body>
<c:if test="true">
무조건 수행<br>
</c:if>

<c:if test="${param.name == 'kkh'}">
name 파라미터의 값이 ${param.name} 입니다.<br>
</c:if>

<c:if test="${18 < param.age}">
당신의 나이는 18세 이상입니다.
</c:if>
</body>
</html>
```

결과



- <c:choose> tag

choose 태그는 자바의 switch 구문과 if-else 블록을 혼합한 형태로서 다수의 조건문을 하나의 블록에서 수행하고자할 때 사용된다.

choose 태그는 다수의 when tag 가 중첩되어있는데 각각의 when 태그는 test 속성의 결과값이 true 일경우 내부 블록을 수행한다.

만약 모든 when 태그의 조건결과값이 false 였다면 otherwise 태그가 실행된다.

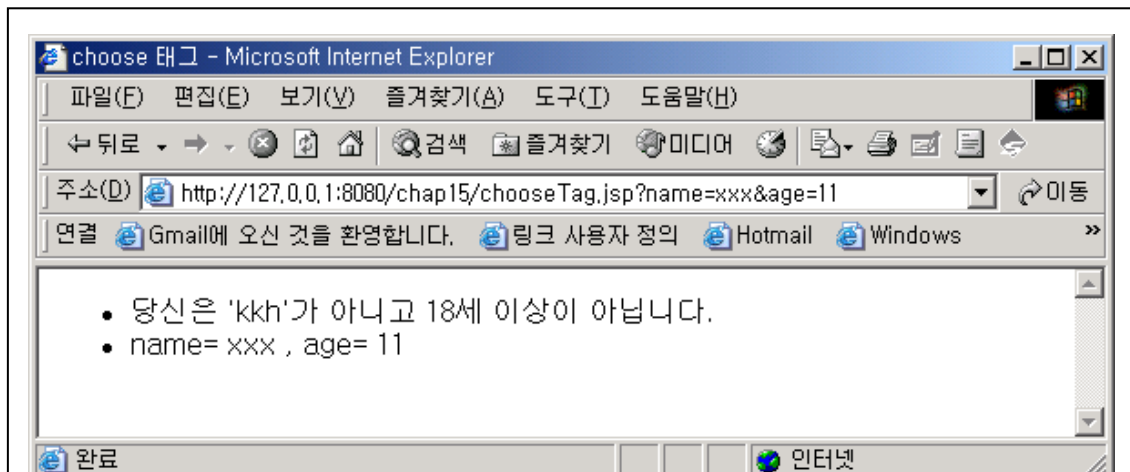
형식 :

```
<c:choose >
    <c:when test="조건1">
        문장1..
    </c:when>
    <c:when test="조건2">
        문장2..
    </c:when>
    <c:when test="조건3">
        문장3..
    </c:when>
    <c:otherwise>
        문장4..
    </c:otherwise>
</c:choose>
```

chooseTest.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head><title>choose 태그</title></head>
<body>
<ul>
<c:choose>
  <c:when test="${param.name == 'kkh'}" >
    <li> 당신의 이름은 ${param.name} 입니다.
  </c:when>
  <c:when test="${param.age > 18}" >
    <li> 당신은 18세 이상입니다.
  </c:when>
  <c:otherwise>
    <li> 당신은 'kkh'가 아니고 18세 이상이 아닙니다.
    <li> name= ${param.name} , age= ${param.age}
  </c:otherwise>
</c:choose>
</ul>
</body>
</html>
```

결과





## - <c:forEach> tag

forEach 태그는 배열, Collection, 또는 Map 에 저장되어있는 값들을 순차적으로 처리할때 사용할 수 있는 태그로서, 자바의 for, do-while 등을 대신해서 사용할 수 있는 태그이다.

형식 :

```
<c:forEach var="변수" items="아이템">
    ${변수사용}
</c:forEach>
```

forEachTag.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%
    java.util.HashMap mapData = new java.util.HashMap();
    mapData.put("name", "김경호");
    mapData.put("today", new java.util.Date());
%>
<c:set var="intArray" value="<%= new int[] {1,2,3,4,5} %>" />
<c:set var="map" value="<%= mapData %>" />
<html>
<head><title>forEach 태그</title></head>
<body>
<h4>1부터 100까지 홀수의 합</h4>
<c:set var="sum" value="0" />
<c:forEach var="i" begin="1" end="100" step="2">
    <c:set var="sum" value="${sum + i}" />
</c:forEach>
결과 = ${sum}

<h4>구구단: 4단</h4>
<ul>
<c:forEach var="i" begin="1" end="9">
```

```
<li>4 * ${i} = ${4 * i}
</c:forEach>
</ul>
```

```
<h4>int형 배열</h4>
```

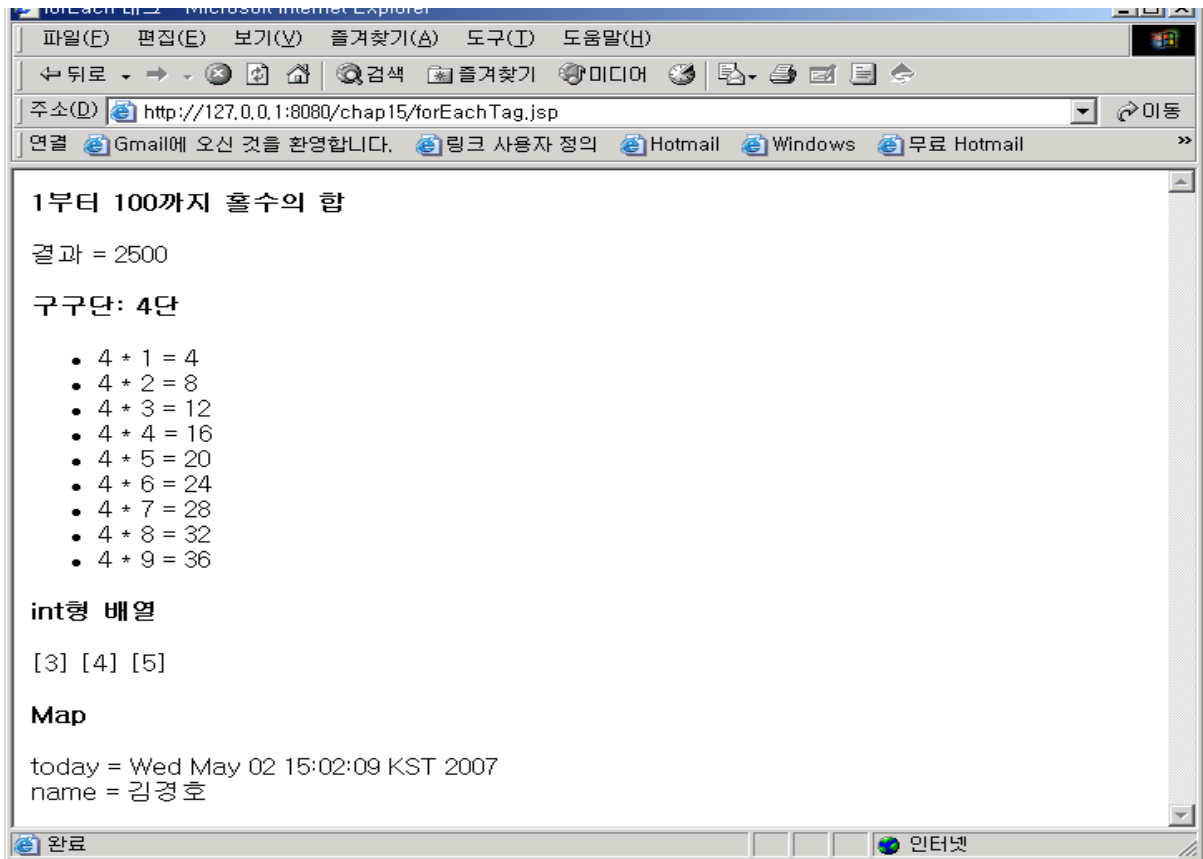
```
<c:forEach var="object" items="${intArray}" begin="2" end="4">
    [ ${object } ]
</c:forEach>
```

```
<h4>Map</h4>
```

```
<c:forEach var="object" items="${map}">
    ${ object.key} = ${ object.value}<br>
</c:forEach>
</body>
</html>
```

---

결과



## -URL 처리태그

import, redirect, url 등의 태그가 있다.

### -<c:import>태그

import 태그는 <jsp:include>태그와 비슷한기능을 제공하는 태그이다

형식 :

```
<c:import url="http://media.daum.net" />
  charEncoding="euc-kr"
  var="daumNews"
  scope="request"
  <c:param name="_top_G" value="news"/>
</c:import>
```

- url -- 읽어올 URL
- charEncoding -- 읽어 온 데이터의 캐릭터셋

- var - 읽어온 데이터를 저장할 변수명
- scope - 변수를 저장할 범위지정

#### -<c:url>태그

url 태그는 URL 을 생성해서 변수에 저장할 때 사용되며 사용방법은 다음과 같다.

#### 형식 :

```
<c:url var="url2" value="/shopping.do" >
  <c:param name="Add" value="isdn-001" />
</c:url>
```

url 태그의 각 속성은 다음과 같다.

- var - 생성한 URL 이 저장될 변수명
- value - 생성할 변수명
- scope - 변수를 저장할 범위지정

#### -<c:redirect>태그

redirect 태그는 지정한 페이지로 redirect 할 때 사용되며  
response.sendRedirect()와 비슷한 기능을 가짐  
사용방법은 다음과 같다.

#### 형식 :

```
<c:redirect url="ifTag.jsp" >
  <c:param name="name" value="kh" />
</c:redirect>
```

포함되는 파일

importTag.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:import url="/url.jsp" var="urlEx" />
<c:import url="http://localhost:8080/chap15/header.jsp" var="head">
    <c:param name="id" value="kkh" />
</c:import>
<c:import url="/footer.jsp" var="foot" />

<html>
<head><title>import와 url 태그</title></head>
<body>
${head}
${urlEx}
${foot}
</body>
</html>
```

urlTag.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:url var="url1" value="../shopping.do" />
<c:url var="url2" value="/shopping.do" >
    <c:param name="Add" value="isdn-001" />
</c:url>
<c:url var="url3" value="http://localhost:8080/jstl/setTag.jsp" />
<ul>
    <li>URL1 = ${url1}
    <li>URL2 = ${url2}
    <li>URL3 = ${url3}
</ul>
```

header.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<hr size="1">
<c:if test="${param.id != null}">
[HEADER] ${param.id} 님의 방문을 환영합니다.
<hr size="1">
</c:if>
```

footer.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<hr size="1">
<center>footer area</center>
```

## - 기타 태그

out, catch 등의 태그가 있다.

### -<c:out> 태그

out 태그는 JspWriter 에 데이터를 출력할 때 사용하는 태그이다.

형식 :

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

out 태그의 각 속성은 다음과 같다.

- Value - 문자열 혹은 Reader(데이터를 읽어서 출력)
- EscapeXml - 이 속성의 값이 true 일 경우 <, >, &, ', " 이런 문자를 &lt;, &gt;, &amp;, ', " 로 변환한다

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ page import = "java.io.IOException, java.io.FileReader" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head><title>소스 보기</title></head>
<body>
<%
    FileReader reader = null;
    try {
        String path = request.getParameter("path");
        reader = new FileReader(getServletContext().getRealPath(path));
    %>
<pre>
소스 코드 = <%= path %>
<c:out value="<%= reader %>" escapeXml="true" />
</pre>
<%    } catch(IOException ex) { %>
에러: <%= ex.getMessage() %>
<%
    } finally {
        if (reader != null)
            try {
                reader.close();
            } catch(IOException ex) {}
    }
%>
```