

액션태그

JSP



Action Tag

태그의 종류	설명
<jsp:forward>	다른 사이트로 이동할 때 사용한다. 페이지의 흐름을 제어할 때 사용한다.
<jsp:include>	정적 혹은 동적인 자원을 현재 페이지의 내용에 포함시킨다. 페이지를 모듈화할 때 사용한다.
<jsp:param>	<jsp:forward>, <jsp:include>, <jsp:plugin>과 같이 사용되어 파라미터를 추가할 때 사용한다.
<jsp:useBean>	빈(Beans)을 생성하고 사용하기 위한 환경을 정의하는 액션 태그이다.
<jsp:setProperty>	액션은 빈에서 속성 값을 할당한다.
<jsp:getProperty>	액션은 빈에서 속성 값을 얻어올 때 사용한다.

Action Tag

- ❖ Action Tag는 XML문법을 따른다.
- ❖ 시작태그와 종료태그 사용
- ❖ Action Tag 형식

`<jsp: ... 속성="값"> 내용 </jsp:...>`

ex)

```
<jsp:forward page="yellow.jsp">  
    <jsp:param name="url" value="red.jsp"/>  
    <jsp:param name="news" value="Hello, Action Tag!"/>  
</jsp:forward>
```

Action Tag

❖ Action Tag에 속성 값만 지정하고 내용이 없는 경우 XML규칙에 의해 끝나는 태그를 따로 하지 않고 시작태그의 마지막 부분을 “/>”로 닫는다.

❖ 형식

- `<jsp:… 속성=“값”/>`

ex)

```
<jsp:forward page=“yellow.jsp”/>
```

<jsp:forward>

- ❖ 현재 JSP 페이지에서 URL로 지정한 특정 페이지로 넘어갈 때 사용

```
<jsp:forward page="이동 할 페이지">
```

```
<jsp:param name="변수명" value="값"/>
```

```
</jsp:forward>
```

ex)

```
<jsp:forward page="relativeURLspec"/>
```

- ❖ 아래의 기존 forward() 메서드를 사용하는 방식보다 액션 태그를 사용하는 것이 소스 길이가 간단하고 가독성이 높은 페이지를 만들 수 있다.

- ❖ EX01_select.jsp, EX01_view.jsp, a.jsp, b.jsp, c.jsp 실습

```
RequestDispatcher dispatcher = request.getRequestDispatcher("url");  
dispatcher.forward(request, response);
```

<jsp:include>

❖ 다른 jsp페이지를 현재 페이지에 삽입하는 태그

- 웹 사이트의 페이지 상단에는 로고나 메인 메뉴, 페이지 하단의 저작권 표시 등과 같이 모든 페이지에서 **공통적으로 사용 되는 부분**이 있다.
- 이런 공통된 내용까지 하나의 JSP 페이지에 기술하는 것 보다 페이지 상단의 로고나 메인 메뉴는 header.jsp에 페이지 하단의 저작권 표시 등은 footer.jsp에 나누어 두고, 본문에 해당되는 내용을 기술하는 페이지에서 이들 header.jsp와 footer.jsp를 동적으로 포함하는 것이 좋다.
- header.jsp와 footer.jsp를 포함시킬 때 <jsp:include> 액션 태그를 사용한다.

include Directive & include Actiontag

❖ 두 태그는 같은 기능을 하지만 동작원리가 다르다.

❖ **include** 지시자 : `<%@ include file= 'url' %>`

지정된 페이지가 현재 페이지에 포함되어 하나로 합쳐진 상태에서 컴파일하여 실행되는 구조, 포함되는 페이지를 독립적인 형태가 아니라 페이지의 일부분으로 구성한다. 즉, 변수를 서로 공유해서 사용할 수 있다.

❖ **include** 액션 태그 : `<jsp:include>`

컴파일할 때 두 페이지가 하나로 합쳐지지 않고, 제어권이 지정한 페이지로 넘어갔다가 다시 되돌아온다. 서로 독립적으로 컴파일되고 실행되기 때문에 변수를 서로 공유할 수 없다.

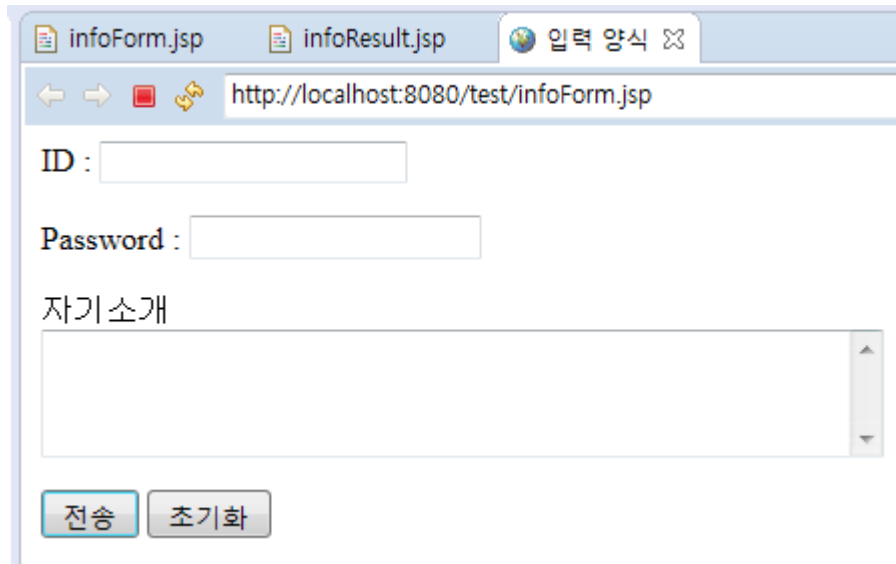
❖ 실습

- top.jsp, bottom.jsp, left.jsp
- template.jsp, info_view.jsp, infoSub.jsp, mainPage.jsp

예제[1]

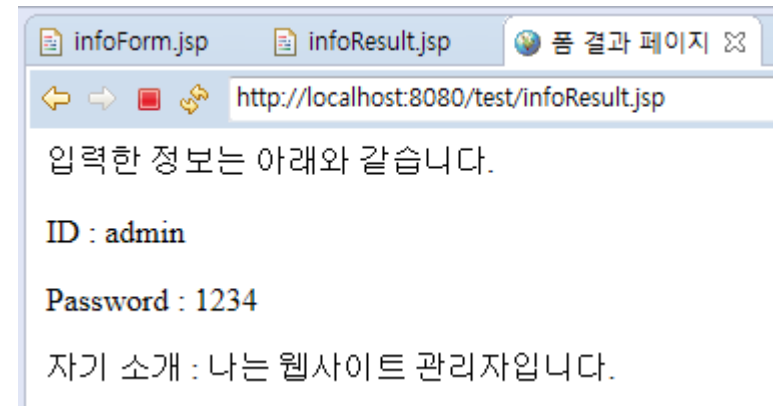
- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 결과값은 request.getParameter()를 사용하세요.

<infoForm.jsp : 입력 양식 페이지>



The screenshot shows a web browser window with two tabs: 'infoForm.jsp' and 'infoResult.jsp'. The active tab is 'infoForm.jsp', which displays a form titled '입력 양식' (Input Form). The form contains three input fields: 'ID :', 'Password :', and '자기소개' (Self-introduction). Below the '자기소개' field is a large text area. At the bottom of the form are two buttons: '전송' (Submit) and '초기화' (Reset). The browser's address bar shows the URL 'http://localhost:8080/test/infoForm.jsp'.

<infoResult.jsp : 결과 페이지>



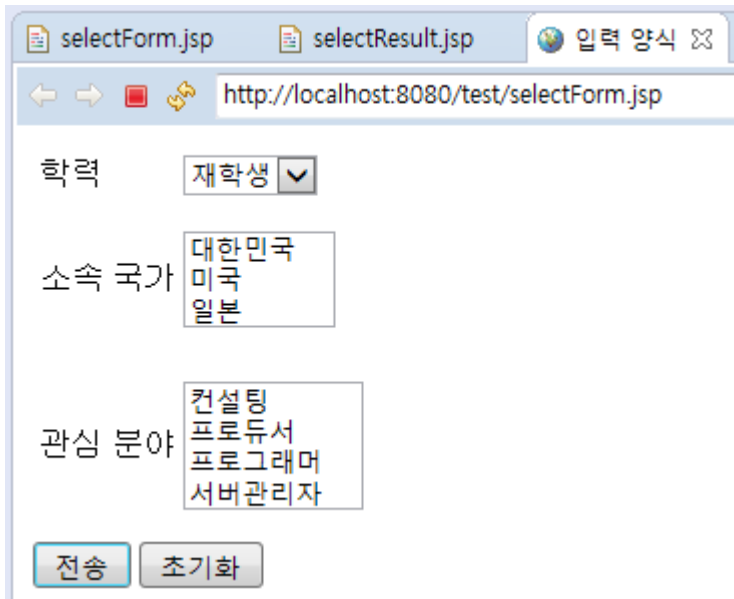
The screenshot shows a web browser window with two tabs: 'infoForm.jsp' and 'infoResult.jsp'. The active tab is 'infoResult.jsp', which displays a page titled '폼 결과 페이지' (Form Result Page). The page shows the submitted information: '입력한 정보는 아래와 같습니다.' (The entered information is as follows.), 'ID : admin', 'Password : 1234', and '자기 소개 : 나는 웹사이트 관리자입니다.' (Self-introduction : I am a website administrator.). The browser's address bar shows the URL 'http://localhost:8080/test/infoResult.jsp'.

예제[2]

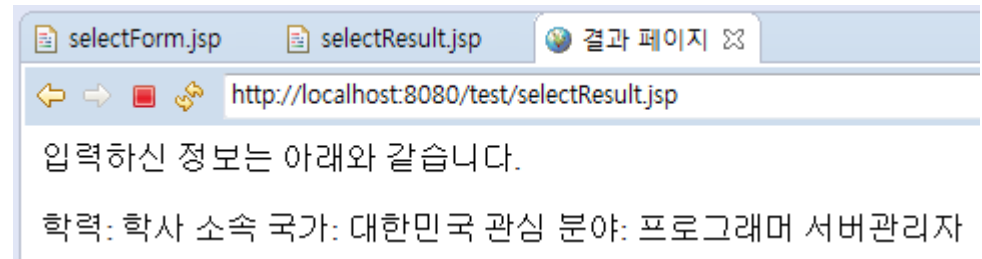
- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 결과값은 request.getParameter()와 request.getParameterValues(여러 개의 값)를 사용하세요.
- 입력 페이지에서 관심분야는 다중 선택(mutiple)이 가능하도록 하고, 결과 페이지에서 관심분야를 출력할 때는 배열 변수와 for문을 사용하세요.

<selectForm.jsp : 입력 양식 페이지>

<selectResult.jsp : 결과 페이지>



The screenshot shows a web browser window with two tabs: 'selectForm.jsp' and 'selectResult.jsp'. The active tab is 'selectForm.jsp' at the URL 'http://localhost:8080/test/selectForm.jsp'. The page contains three form fields: '학력' (Education) with a dropdown menu showing '재학생' (Current Student); '소속 국가' (Affiliated Country) with a list box showing '대한민국' (South Korea), '미국' (USA), and '일본' (Japan); and '관심 분야' (Area of Interest) with a list box showing '컨설팅' (Consulting), '프로듀서' (Producer), '프로그래머' (Programmer), and '서버관리자' (Server Administrator). At the bottom, there are two buttons: '전송' (Submit) and '초기화' (Reset).

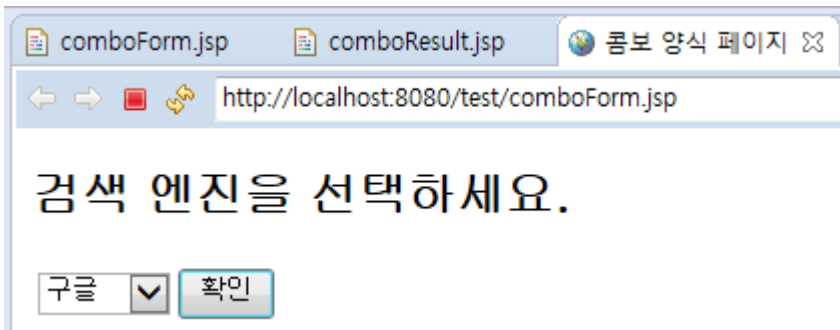


The screenshot shows a web browser window with two tabs: 'selectForm.jsp' and 'selectResult.jsp'. The active tab is 'selectResult.jsp' at the URL 'http://localhost:8080/test/selectResult.jsp'. The page displays the message '입력하신 정보는 아래와 같습니다.' (The information you entered is as follows.) followed by the text '학력: 학사 소속 국가: 대한민국 관심 분야: 프로그래머 서버관리자' (Education: Bachelor's Degree Affiliated Country: South Korea Area of Interest: Programmer Server Administrator).

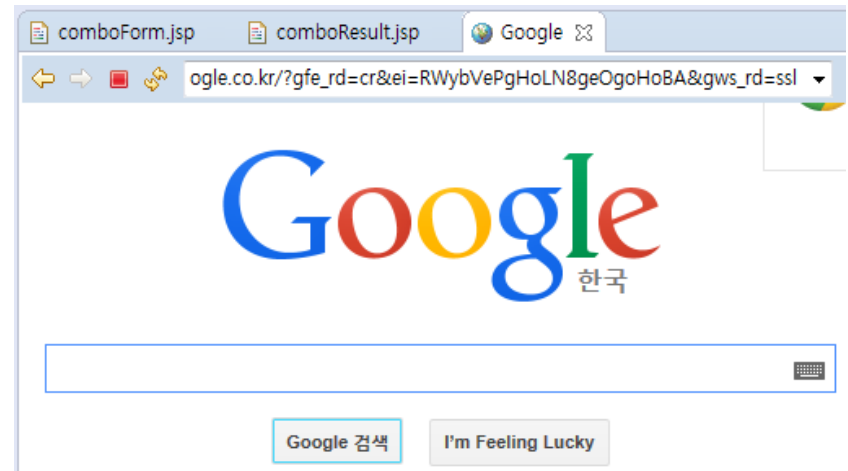
예제(3)

- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 콤보 박스에서 항목을 선택하면 해당 사이트로 이동되도록 만드세요.
- 결과값은 request.getParameter()를 사용하세요.

<comboForm.jsp : 콤보 양식 페이지>



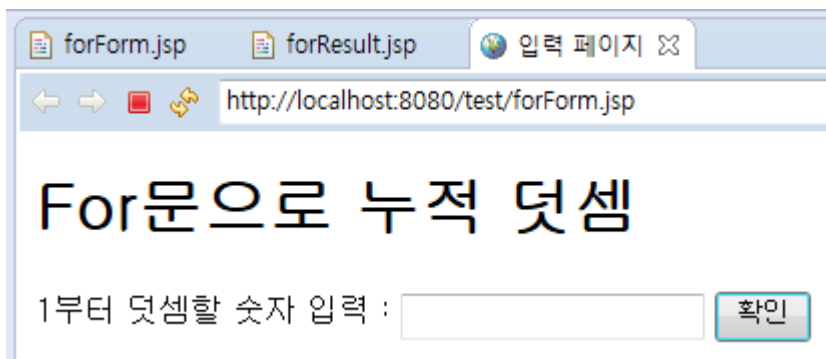
<comboResult.jsp : 결과 페이지>



예제[4]

- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 입력 페이지에서 정수를 입력하고 [확인] 버튼을 누르면, 결과 페이지에서는 1부터 입력한 숫자까지의 합계가 출력되도록 만드세요.
- 결과 페이지에서 합계를 구하는 함수를 선언하여 사용하고, 결과값은 request.getParameter()와 for문을 사용하여 출력하세요.

<forForm.jsp : 입력 양식 페이지>



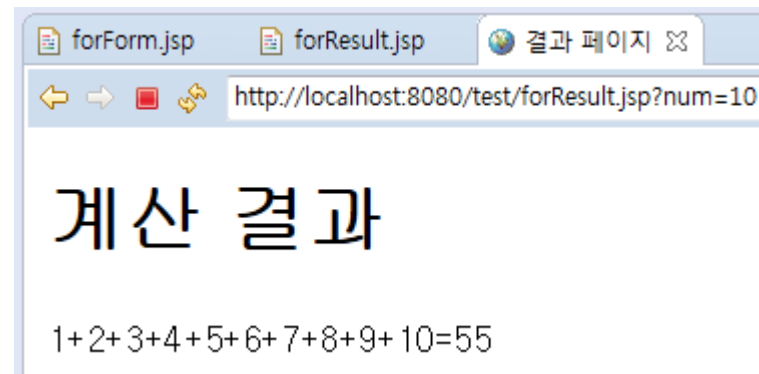
forForm.jsp forResult.jsp 입력 페이지

← → http://localhost:8080/test/forForm.jsp

For문으로 누적 덧셈

1부터 덧셈할 숫자 입력 :

<forResult.jsp : 결과 페이지>



forForm.jsp forResult.jsp 결과 페이지

← → http://localhost:8080/test/forResult.jsp?num=10

계산 결과

1+2+3+4+5+6+7+8+9+10=55

JavaBeans

❖ JavaBeans

- 자바 클래스 중에서 자바빈즈 규약에 맞게 작성된 클래스다.
- 멤버 변수와 get/set 메서드로 이루어져 있다.
- 값을 저장하는 VO로 사용한다.

❖ <jsp:useBean>

- 객체의 이름/사용범위/저장위치 등으로 통해서 객체 생성
- JSP의 자바코드에서 actiond의 id특성에 지정된 값을 통해 객체를 참조한다.

❖ 형식

<jsp:useBean id="Bean 이름" scope="범위" class="Bean의 저장위치"/>

- id : 객체 식별자
- scope : 객체 참조 범위
- class : 완전한 형태의 클래스 이름

setProperty/getProperty

❖ <jsp:setProperty>

- 함께 사용되는 액션태그
- Bean의 속성에 값을 설정하는 태그
- 형식
`<jsp:useBean id="Bean 이름" property="속성이름" value="저장할 값"/>`
- name : <jsp:useBean> 태그에 정의된 Bean 인스턴스 이름
- property : 값을 설정할 Bean 속성이름
(' * '설정 시 ServletRequest안의 모든 인자 중 Bean 속성과 자료형이 일치하는 것을 찾아 모두 지정된다.)
- value : Bean 속성에 설정할 값

❖ <jsp:getProperty>

- Bean의 속성 값을 얻을 때 사용하는 태그
- 형식
`<jsp:getProperty name="Bean 이름" property="속성 이름" />`
- name : 얻을 Bean객체 이름
- property : 가져올 속성의 이름