

## 05. 액션태그 - 자바빈즈

### 1) 개요

- : 비즈니스로직을 담고 있으며, 자바빈즈 규약을 따르는 클래스
- : 특정 데이터를 관리하고 표현하기 위해 JSP 에서 사용하는 특수한 형태의 클래스

### 2) 빈즈 클래스의 구성이자 조건

(0) 구성 :

- ① 생성자
- ② 멤버변수
- ③ 데이터 저장을 위한 set 형태의 멤버 method
- ④ 데이터 추출을 위한 get 형태의 멤버 method

(1) 반드시 **public class** 로 작성해야함. 누구나 접근할 수 있어야 하기 때문

➔ 웹상에서 불러다 사용할 수 있도록

➔ 메서드를 호출하기 위해서

(2) **멤버변수(입력받는 부분)**은 반드시 **캡슐화**하여야 한다. => **private**

(3) **Setter, Getter Method** 작성

### 3) 자바빈즈 3 가지 태그

#### 3-1) useBean

: 이 Bean 을 use 할거야 ➔ **Bean 객체 생성**

형식)

```
<jsp:useBean id="객체명" class="패키지명.클래스명" scope="사용범위"/>
```

#### 3-2) setProperty

: 선언된 객체의 set 멤버 method 를 호출하여 **변수에 값을 설정**하는 태그

형식)

```
<jsp:setProperty name="객체명" property="멤버변수명(str)" value="저장할 값"/>
```

### 3-3) getProperty

: 선언된 객체의 get 멤버 method 를 호출하여 **변수 값을 리턴**하는 태그

형식)

```
<jsp:getProperty name="객체명" property="멤버변수명(str)" />
```

### 4) 자바코드를 작성하는 경우 VS 액션태그

회원가입을 할 때, 사용자의 정보를 20 개 가량을 받는다고 가정하자

#### (1) 자바코드를 사용한다면

```
String str=request.getParameter("str");
String addr=request.getParameter("addr");
....

BeanTest bt=new BeanTest();
bt.setStr(str);
,,,=====>20줄
```

각각의 정보를 일일이 입력해줘야한다.

#### (2) 액션태그(자바빈즈) 를 사용한다면

```
<jsp:setProperty name="객체명" property="*" />
```

위의 한줄로 간결하게 끝낼 수 있다.

## beans.jsp : 자바빈즈가 나오게 된 배경

```
<body>
```

```
<%! → 선언문형태
```

```
// 입력을 받아서 > 매개변수를 전달 > 각 값을 필드별로 저장하는 변수(멤버변수(저장 테이블의 필드에 저장)
```

```
String str = "선언문"; //웹상에서 입력받을 값이 저장되는 멤버변수
```

```
String addr = "";
```

```
//Web에서는 생성자(Constructor)로 값을 넣지 않고, Setter, Getter를 이용한다.
```

```
// Java 구문과 화면구현 구문이 한 문서에 섞이면
```

```
// 1. 화면 디자인이 어렵다 (복잡)
```

```
// 2. 기존의 코딩을 다른 사이트에 적용하기가 어렵다.(재사용성이 떨어진다)
```

```
public void setStr(String s){ // 입력-> 저장
```

```
    str = s;
```

```
    System.out.println("setStr()호출");
```

```
}
```

```
public void setAddr(String a){
```

```
    addr = a;
```

```
    System.out.println("setAddr()호출");
```

```
}
```

```
public String getStr(){
```

```
    return str;
```

```
}
```

```
public String getAddr(){
```

```
    return addr;
```

```
}
```

```
%>
```

```
<p>메서드 호출 (Setter Method):<% setStr("자바빈즈"); %></p>
```

```
<p>저장된 값 출력(Getter Method):<h1><%=getStr() %></h1>
```

```
</body>
```

```
setStr 메소드로 문자열 설정  
getStr() 메소드로 문자열 얻어옴
```

전체는 html 코드  
(jsp파일)

## form.jsp

: 데이터를 입력받고, 그 값을 beans2.jsp 로 전달한 뒤 그 값을 BeanDTO 객체에 담도록

```
<body>
<h1>include 액션태그 연습</h1>
<form action="beans2.jsp" method="post">
  <p>이름: <input type="text" name="name" /></p>
  <p>주소: <input type="text" name="addr" /></p>
  <!-- <p>나이: <input type="text" name="age" /></p> -->
  <p><input type="submit" value="전송" /></p>
</form>

</body>
```

## beans.jsp : 빈즈클래스를 javacode 로 활용

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="test.BeanDTO" %>
<!DOCTYPE html>
<html>
<body>
<%
    request.setCharacterEncoding("utf-8"); // 한글데이터 처리
    String name = request.getParameter("name");
    String addr = request.getParameter("addr");
    // 1. 객체생성 (메소드 호출, 데이터 저장)
    BeanDTO bd = new BeanDTO();
    // 2. 입력받은 갯수만큼 Setter Method 호출 => 데이터 값을 저장한다.
    bd.setName(name); // bd.setName(request.getParameter("name")); → 값을 넘겨받는다면
    bd.setAddr(addr);
    // 3. Getter Method를 호출해서 화면에출력
    out.println("입력받은 이름: "+bd.getName()+"<br>");
    out.println("입력받은 주소: "+bd.getAddr()+"<br>");
%>
</body>
</html>
```

입력받은 값 전달

BeanDTO객체 bd 생성

Java Code

객체의 get 메소드로 값 가져옴

## form2.jsp

: 데이터를 입력받고, 그 값을 beans3.jsp 로 전달한 뒤 그 값을 BeanDTO 객체에 담도록

```
<body>
<h1>include 액션태그 연습</h1>
<form action="beans3.jsp" method="post">
  <p>이름: <input type="text" name="name"> </p>
  <p>주소: <input type="text" name="addr"> </p>
  <!-- <p>나이: <input type="text" name="age"> </p> -->
  <p><input type="submit" value="전송"> </p>
</form>

</body>
```

## beans3.jsp : 빈즈클래스를 javacode 로 활용

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="test.BeanDTO" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<%
```

```
    request.setCharacterEncoding("utf-8"); // 한글데이터 처리
```

```
    String name = request.getParameter("name");
```

```
    String addr = request.getParameter("addr");
```

입력받은 값 전달

BeanDTO객체 bd 생성

```
%>
```

```
<jsp:useBean id="bd" class="test.BeanDTO" scope="page"/>
```

```
<%--
```

```
<jsp:setProperty name="bd" property="name" value="<%=name %>"/> → 주석처리 됨
```

```
<jsp:setProperty name="bd" property="addr" value="<%=addr %>"/> → 주석처리 됨
```

```
--%>
```

```
<jsp:setProperty name="bd" property="*" />
```

```
<jsp:getProperty name="bd" property="name" />
```

```
<jsp:getProperty name="bd" property="addr" />
```

```
</body>
```

```
</html>
```

setProperty로 각각의 값들을 입력하기엔 번거롭기 때문에 \* (Asterisk) 를 사용해서 "입력받은 순서대로" 모두 가져온다. 메모리에 임시로 가져오는게 아닌 서버에 저장하는 기능이다.

getProperty로 필요한 값들을 가져와서 보여줄 수 있다.

Java Code

## ※ 빈즈를 이용한 계산기 구현

### cal.jsp: 두개의 정수값을 입력받고, 계산

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="calc.CalcBean"%>
<jsp:useBean id="ca" class="calc.CalcBean" scope="request"/>
<jsp:setProperty name="ca" property="*" />
<% ca.calculate(); %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>요청을 하는 페이지(빈즈이용)</title>
</head>
<body style="text-align:center;">
<h3> 간단한 계산기(요청, 응답을 하나의 페이지에서 작성 Ajax)</h3>
<!-- <form method="post" action="cal.jsp" name="form1"> -->
<form name="form1" method="post">
    <p>
        <input type="text" name="num1" width="200" size="5">
        <select name="operator">
            <option value="+" selected>+</option>
            <option value="-">-</option>
            <option value="*">*</option>
            <option value="/">/</option>
        </select>
        <input type="text" name="num2" width="200" size="5">
    </p>
    <input type="submit" value="계산">
    <input type="reset" value="다시입력">
</form>
<hr>
계산결과 <jsp:getProperty name="ca" property="result"/>
</body>
```

①

1. CalcBean의 객체 ca를 만들고 초기값을 셋팅한다. (모두 0)  
2. ca의 calculate() method를 실행하여 값이 반환된 상태(0)로 html문을 시작한다.

③

1. form에 입력한 값이 setProperty 를 통해 ca 객체에 입력되고, ca의 calculate() method를 실행하여 계산 결과가 반환된다.

따로 **action**을 표시해주지 않으면 **자기 자신으로 보낸다.**

②

값을 입력하고 submit을 누르면 다시 이 문서의 처음으로 돌아간다