

2장. JSP 태그



JSP Tag



JSP 태그 – 스크립트 태그

JSP 태그

- **스크립트 태그** : HTML 코드에 자바 코드를 넣어 프로그램을 수행하는 기능
- **디렉티브 태그** : 현재 JSP 페이지에 대한 정보를 설정하는 태그
- **액션 태그** : 클라이언트/서버에 어떤 작동을 행하도록 명령을 지시하는 태그

스크립트 태그(script tag)

스크립트 태그	형식	설 명
선언문(declaration)	<%! %>	자바 변수나 메소드를 정의하는 데 사용
스크립트릿(scriptlet)	<% %>	자바 로직 코드를 작성하는 데 사용
표현문(expression)	<%= %>	변수, 메소드 호출 결과를 문자열로 출력해 줌



JSP 태그 - 스크립트 태그

구글 - JSP Document - 검색

<https://docs.oracle.com/javaee/5/tutorial/doc/>

The Java EE 5 Tutorial

Table 6-1 Standard Syntax Versus XML Syntax

Syntax Elements	Standard Syntax	XML Syntax
Comments	<code><%-- .. --%></code>	<code><!-- .. --></code>
Declarations	<code><%! ..%></code>	<code><jsp:declaration> .. </jsp:declaration></code>
Directives	<code><%@ include .. %></code>	<code><jsp:directive.include .. /></code>
	<code><%@ page .. %></code>	<code><jsp:directive.page .. /></code>
	<code><%@ taglib .. %></code>	<code>xmlns:prefix="tag library URL"</code>
Expressions	<code><%= ..%></code>	<code><jsp:expression> .. </jsp:expression></code>
Scriptlets	<code><% ..%></code>	<code><jsp:scriptlet> .. </jsp:scriptlet></code>



JSP 태그 – 스크립트 태그

선언문과 스크립틀릿 태그

```
<body>
  <h2>Scripting Tag</h2>
  <!-- 선언문 태그 -->
  <%!
    int num1 = 10, num2 = 20;
    int sum = 0;
  %>
  <%-- 스트립틀릿 태그 --%>
  <%
    //더하기
    sum = num1 + num2;
    out.println("sum = " + sum + "<br>");

    //짝수 출력
    for(int i = 1; i <= 10; i++){
      if(i % 2 == 0)
        out.println(i);
    }
  %>
</body>
```

scripting.jsp

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Scripting Tag</title>
</head>
<body>
  <h2>Scripting Tag</h2>

  sum = 30<br>
  2
  4
  6
  8
  10

</body>
</html>
```

소스보기 – jsp 태그는 서버에서 처리되므로 클라이언트(웹 브라우저)에서는 보이지 않음



JSP 태그 – 스크립트 태그

표현문(expression) 태그

웹 브라우저에 출력할 부분 표현

expression.jsp

```
<title>표현문 태그</title>
</head>
<!-- 전역 변수 선언 -->
<%!
    int count = 0;
%>
<body>

    <h3>Page Count is<%= ++count%></h3>

    <p>Today's date: <%= new java.util.Date().toLocaleString() %></p>

</body>
</html>
```

← → ↻ ⓘ localhost:8080/jwbook/ch02/expression.jsp

Page Count is 2

Today's date: 2022. 7. 16. 오전 5:57:59



JSP 태그 – 스크립트 태그

- 실습 예제

실행 결과 대로 JSP 애플리케이션을 만들어 보세요.

표현문 태그로 출력해야 하고, 파일이름은 expression02.jsp

홀수/짝수를 판별하는 프로그램

결과는 홀수입니다.



JSP 태그 – 디렉티브 태그

디렉티브 태그

- JSP 페이지를 어떻게 처리할 것인지를 설정하는 태그
- JSP 페이지가 서블릿 프로그램에서 서블릿 클래스로 변환할 때 JSP 페이지와 관련된 정보를 JSP 컨테이너에 지시하는 메시지 이다.

디렉티브 태그의 종류

디렉티브 태그	형식	설 명
page	<code><%@ page ... %></code>	JSP 페이지에 대한 정보를 설정
include	<code><%@ include ... %></code>	JSP 페이지의 특정 영역에 다른 문서 포함
taglib	<code><%@ taglib ... %></code>	JSP 페이지에 사용할 태그 라이브러리 설정



JSP 태그 – 디렉티브 태그

page 디렉티브 태그 – contentType 속성

- 현재 JSP 페이지에 대한 정보를 설정하는 태그
- 콘텐츠 문서의 유형, 사용할 자바 클래스, 오류 페이지 설정, 세션 사용 여부 등

`<%@ page contentType = "문서의 유형" %>`

The image shows a web browser window and its source code. The browser window on the left displays the text "Welcome" and "í™~ì~□í•©ë<^ë<α." which is garbled text. The source code in the middle shows the JSP page structure with the `<%@ page contentType="text/html; charset=utf-8" %>` tag highlighted in a red box. The browser window on the right displays the text "Welcome" and "환영합니다." which is the correct rendering of the page.

```
<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>Welcome</title>
</head>
<body>
  <h1>Welcome</h1>
  <h1>환영합니다.</h1>
</body>
</html>
```



JSP 태그 – 디렉티브 태그

page 디렉티브 태그 – import

<%@ page import = “자바클래스” %>

```
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.Arrays"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>페이지 디렉티브 태그</title>
</head>
<body>
    <%
        //배열
        int[] arr = new int[]{10, 20, 30};
        for(int i = 0; i < arr.length; i++){
            out.println(arr[i]);
        }

        out.println("<br>");
    %>
</body>
</html>
```

import.jsp



JSP 태그 – 디렉티브 태그

page 디렉티브 태그

```
//Arrayss 클래스
int[] arr2 = new int[]{10, 20, 30};
out.println(Arrays.toString(arr2));

//ArrayList 클래스
ArrayList<String> fruitList = new ArrayList<>();
fruitList.add("사과");
fruitList.add("참외");
fruitList.add("수박");

out.println("<br>");
out.println(fruitList.get(1));

out.println("<br>");
for(String fruit : fruitList)
    out.println(fruit);

%>
</body>
</html>
```

← → ↻ ⓘ localhost:8080/jwbook/ch02/import.jsp

10 20 30
[10, 20, 30]
참외
사과 참외 수박



JSP 태그 – 디렉티브 태그

page 디렉티브 태그 – session 속성

```
<%@ page session="true" %>
```

세션은 일반적으로 웹 애플리케이션이 실행되는 동안 사용자가 웹 애플리케이션의 데이터를 가져와 확인할 수 있는 권한을 부여받기 위해 사용

예) 사용자가 쇼핑몰에 로그인하여 로그아웃 될때까지 모든 데이터 및 페이지에 접근할 수 있다.

page 디렉티브 태그 – buffer 속성

```
<%@ page buffer="32KB" %>
```

현재 JSP 페이지 출력 버퍼 크기를 설정하는 데 사용.

기본 버퍼 크기는 서버의 고유한 값으로 8KB이다. 출력 버퍼에 내용이 최소한 8KB 누적될 때까지 웹 브라우저에 전송되지 않는다.



디렉티브 태그

page 디렉티브 태그 – errorPage 속성

```
<%@ page errorPage="errorPage.jsp" %>
```

errorPage 속성은 현재 JSP 페이지가 실행되는 동안 오류가 발생하면 특정 오류 페이지로 이동하는 데 사용.

```
<%@ page errorPage="errorPage.jsp" %>
<!DOCTYPE html>
<html>
<head>
<title>>Directives Tag</title>
</head>
<body>
    <%
        String str = null;
        out.println(str.toString());
    %>
</body>
</html>
```

errorPage_error.jsp

HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

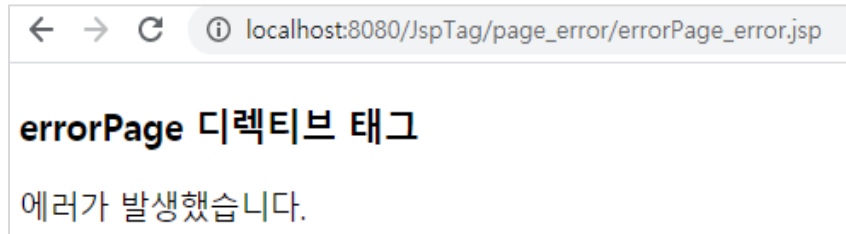
메시지 행 [12]에서 [/errorPageTest.jsp]을(를) 처리하는 중 예외 발생

설명 서버가, 해당 요청을 충족시키지 못하게 하는 예기치 않은 조건을 맞닥뜨렸습니다.



디렉티브 태그

page 디렉티브 태그 – errorPage 속성



```
<%@ page contentType="text/html;
<!DOCTYPE html>
<html>
<head>
<title>Directives Tag</title>
</head>
<body>
    <h3>errorPage 디렉티브 태그</h3>
    에러가 발생했습니다.
</body>
</html>
```

errorPage.jsp

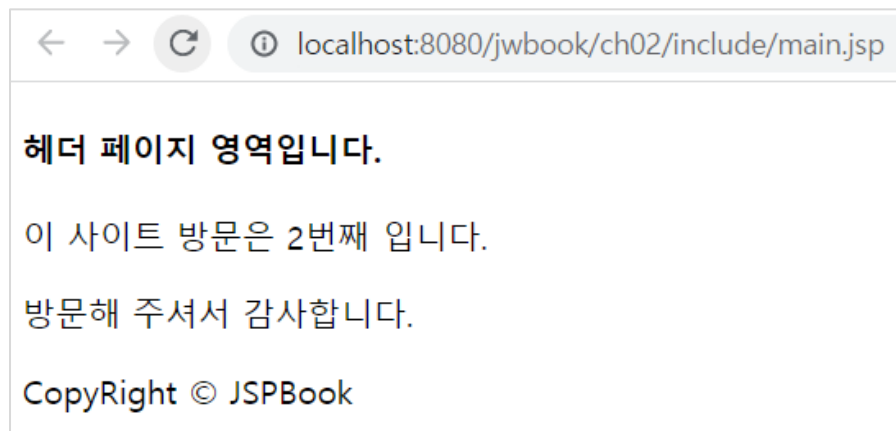


JSP 태그 – 디렉티브 태그

include 디렉티브 태그

<%@ include file="파일명" %>

- 현재 JSP의 특정 영역에 외부 파일의 내용을 포함
- include 디렉티브 태그는 머리글과 바닥글과 같은 공통된 부분을 별도의 JSP 파일을 만들어 모듈화 할 수 있다.



JSP 태그 – 디렉티브 태그

include 디렉티브 태그

```
<body>
    <%!
        int pageCount = 0;

        void addCount(){
            pageCount++;
        }

    %>

    <%
        addCount();
    %>

    <p>이 사이트 방문은 <%=pageCount %>번째 입니다.</p>
</body>
```

include/header.jsp

include/footer.jsp

```
<body>
    <p>CopyRight &copy; JSPBook</p>
</body>
```



JSP 태그 – 디렉티브 태그

include 디렉티브 태그

include/main.jsp

```
<body>
  <%@ include file="header.jsp" %>

  <p>방문해 주셔서 감사합니다.</p>

  <%@ include file="footer.jsp" %>
</body>
```



액션 태그

액션 태그

- 서버나 클라이언트에게 어떤 행동을 하도록 명령하는 태그
- `<% ... %>`와 같은 스크립트 태그 형식이 아닌 **XML 형식** `<jsp: .../>`을 사용함

액션 태그	형식	설 명
forward	<code><jsp:forward ... /></code>	다른 페이지로의 이동과 같은 페이지 흐름 제어
include	<code><jsp:include ... /></code>	외부 페이지의 내용을 포함하거나 페이지를 모듈화
param	<code><jsp:param ... /></code>	<code><jsp:forward></code> , <code><jsp:include></code> 태그에 인자를 추가
useBean	<code><jsp:useBean ... /></code>	JSP 페이지에 자바빈즈를 설정함
setProperty	<code><jsp:setProperty ... /></code>	자바빈즈의 프로퍼티값을 설정함
getProperty	<code><jsp:getProperty ... /></code>	자바빈즈의 프로퍼티값을 얻어옴



XML과 JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<car>
  <brand>Avante</brand>
  <cc>2000</cc>
  <color>Silver</color>
</car>
```

```
<script>
{
  "brand": "Avante",
  "cc": 2000,
  "color": "Silver"
}
</script>
```

XML(eXtensible Markup Language) - 확장될수 있는 표시 언어

Markup Language의 예로는 HTML이 있고, HTML의 태그는 이미 약속한 태그들만 사용 가능하다. XML 태그는 사용자 임의로 만들 수 있고. 필요한 정보들을 받거나 줄 수 있는 데이터 형태를 제공한다.

(사용 예. Jsp web.xml 설정이나 안드로이드 App View 등)

JSON(Javascript object notation) – 자바스크립트 객체 표기

서버에서 클라이언트로 데이터를 보낼 때 사용하는 양식. 클라이언트가 사용하는 언어에 관계 없이 통일된 데이터를 주고받을 수 있음.

XML의 단점(가독성, 데이터 용량)을 보완하여 현재 많이 사용되어 지고 있음



액션 태그

forward 액션 태그

- 현재 JSP page에서 다른 page로 이동하는 태그

```
<jsp:forward page="파일명" />
```

```
<head>
  <title>Action Tag</title>
</head>
<body>
  <h2>forward 액션 태그</h2>

  <jsp:forward page="forward_date.jsp" />

  <p>-----
</body>
</html>
```

action_tag/forward.jsp



액션 태그

```
<%@page import="java.util.Date"%>
<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>Action Tag</title>
</head>
<body>
    <h2>오늘의 날짜 및 시간</h2>

    <p><%=new Date().toLocaleString() %></p>

</body>
</html>
```

action_tag/forward_date.jsp

현재 페이지의 url
이 제어권을 가짐



localhost:8080/jwbook/ch02/action_tag/forward.jsp

오늘의 날짜 및 시간

2022. 7. 17. 오전 7:35:48



액션 태그

param 액션 태그

- 현재 jsp page에서 다른 page에 정보를 전달하는 태그
- 이 태그는 단독으로 사용되지 못하며 <jsp:forward>나 <jsp:include> 태그 내부에 사용됨

```
<jsp:forward page="파일명" >  
    <jsp:param name="매개변수명1" value="매개변수값1" />  
    <jsp:param name="매개변수명2" value="매개변수값2" />  
</jsp:forward>
```

액션 태그

아이디와 이름 전달하기 – forward, param 액션 태그 사용

```
<body>
  <h2>param 액션 태그</h2>
  <jsp:forward page="param1_data.jsp">
    <jsp:param value="admin" name="id"/>
    <jsp:param value="0000" name="pwd"/>
  </jsp:forward>

  <p>Java Server Page</p>
</body>
```

action/param1.jsp

```
<body>
  <p>아이디 : <%=request.getParameter("id") %>
  <p>패스워드 : <%=request.getParameter("pwd") %>
</body>
```

action/param1_data.jsp

← → ↻ ⓘ localhost:8080/jwbook/ch02/action_tag/param1.jsp

아이디: admin

패스워드: 0000



액션 태그

제목(한글)과 현재 날짜 전달하기 – include, param 액션 태그 사용

한글인 경우(인코딩 – 디코딩 클래스)
java.util.URLEncoder -> java.util.Decoder

```
<%@ page import="java.util.Calendar, java.net.URLEncoder"%>
```

action/param2.jsp

```
<h2>param 액션 태그</h2>
<jsp:include page="param2_data.jsp">
  <jsp:param value='<%= URLEncoder.encode("오늘의 날짜와 시각") %>' name="title"/>
  <jsp:param value='<%= Calendar.getInstance().getTime() %>' name="date"/>
</jsp:include>

<!-- 영어인 경우는 인코딩 필요 없음.
      value 값은 반드시 홑따옴표로 감싸야 함(내부 문자열인 경우 쌍따옴표) -->
<!-- <jsp:param value='<%= "Date & Time" %>' name="title"/> --%>
```



액션 태그

```
<%@ page import="java.net.URLDecoder"%>
```

```
<body>  
  <%  
    String title = request.getParameter("title");  
  %>  
  <h3><%=URLDecoder.decode(title) %></h3>  
  <p>Today is : <%=request.getParameter("date") %>  
</body>
```

action/param2_data.jsp

← → ↻ ⓘ localhost:8080/jwbook/ch02/action_tag/param2.jsp

param 액션 태그

오늘의 날짜와 시각

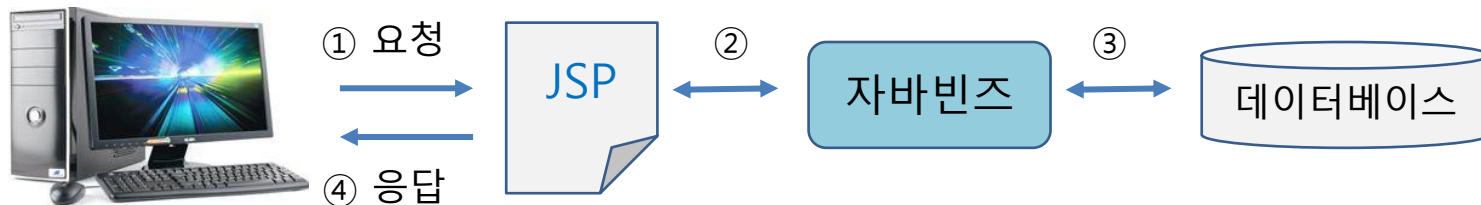
Today is: Sun Jul 17 07:53:02 KST 2022



자바빈즈 액션 태그

자바 빈즈(bbeans)

- 자바 코드를 사용하여 **자바 클래스**로 로직을 작성하는 방법이다.
- 자바코드와 HTML 코드를 함께 작성하면 기능을 확장하거나 코드를 재사용하는데 어려움이 있다.
- JSP 페이지에서 화면을 표현하기 위한 **계산식이나 자료의 처리를 담당하는 자바 코드**를 따로 분리하여 작성한다.



- ① 웹 브라우저가 JSP 페이지에 요청한다.
- ② JSP 페이지는 자바 빈즈와 통신한다.
- ③ 자바 빈즈가 데이터베이스에 연결된다.
- ④ JSP 페이지가 브라우저에 응답한다.

자바빈즈 액션 태그

자바 빈즈 작성 규칙

- ① 자바 클래스는 `java.io.Serializable` 인터페이스를 구현해야 한다.
- ② 인수가 없는 기본 생성자가 있어야 한다.
- ③ 모든 멤버 변수인 프로퍼티는 `private` 접근 지정자로 설정해야 한다.
- ④ 모든 멤버 변수인 프로퍼티는 `Getter/Setter()` 메소드가 존재해야 한다.

```
public class MemberBean implements Serializable{
    private static final long serialVersionUID = 1L;

    private int id = 2021101;
    private String name = "장그래";

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```



자바빈즈 액션 태그

useBean 액션 태그로 자바빈즈 사용하기

- 자바빈즈를 사용하기 위해 실제 자바 클래스를 선언하고 초기화하는 태그.
- 설정된 id, scope 속성으로 자바빈즈의 객체를 검색하고, 객체가 발견되지 않으면 빈 객체를 생성한다.

```
<jsp:useBean id="식별 이름" class="자바빈즈 이름" scope="범위"/>
```

useBean 액션 태그의 속성

속성	설 명
id	자바빈즈를 식별하기 위한 이름이다.
class	패키지 이름을 포함한 자바빈즈의 이름이다.
scope	page(기본값), request, session, application 중 하나의 값을 사용

scope는 생명주기 범위를 이야기하는 것으로 만약 session으로 설정하면 session의 유효 범위 동안 해당 클래스 인스턴스의 상태가 유지된다.



자바빈즈 액션 태그

Calculator 클래스를 생성하여 useBean 액션 태그 사용하기

① 패키지 만들기

src/main/java > 우측 > new > package

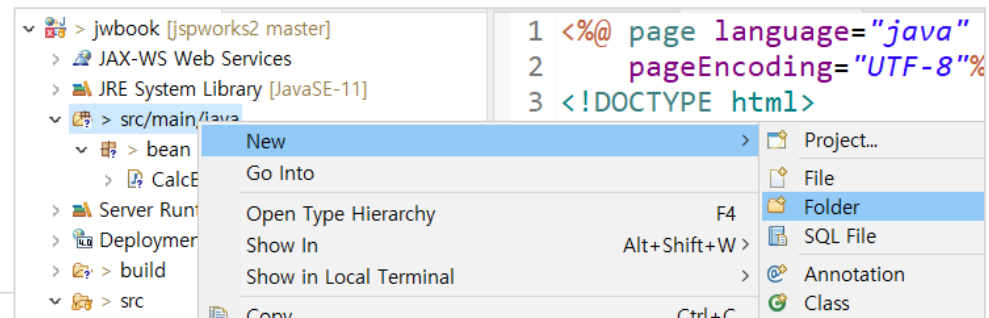
package 이름 : bean

② 클래스 만들기

```
package bean;

public class CalcBean {

    public int calculate(int x) { //세제곱 계산
        return x * x * x;
    }
}
```



자바빈즈 액션 태그

Calculator 클래스를 생성하여 useBean 액션 태그 사용하기

calc/calculator.jsp

```
<body>
  <jsp:useBean id="calc" class="bean.CalcBean" />
  <!-- calc는 CalcBean 클래스의 객체(인스턴스)와 같다. -->
  <%
    int num = calc.calculate(4);
    out.println("4의 3제곱 : " + num);
  %>
</body>
```

← → ↻ ⓘ localhost:8080/jwbook/ch03/calculator.jsp

4의 세제곱: 64



자바빈즈 액션 태그

MemberBean 클래스를 생성하여 useBean 액션 태그로 아이디와 이름 출력

```
public class MemberBean implements Serializable{
    private static final long serialVersionUID = 1L;

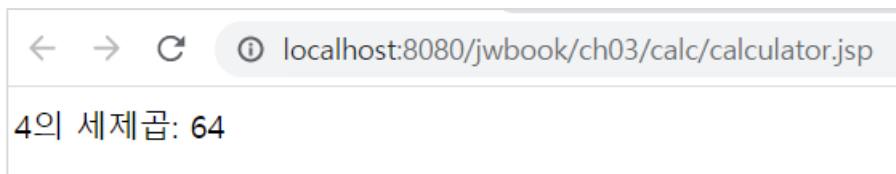
    private int id = 2021101;
    private String name = "장그래";

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```



자바빈즈 액션 태그

MemberBean 클래스를 생성하여 useBean 액션 태그로 아이디와 이름 출력



member/member.jsp

```
<jsp:useBean id="member" class="bean.MemberBean" scope="request"/>
<!-- member는 MemberBean 클래스의 객체(인스턴스)와 같다. -->
<p>아이디 : <%=member.getId() %>
<p>이름 : <%=member.getName() %>
```

request 객체는 저장
기능이 있다.



자바빈즈 액션 태그

getProperty 액션 태그로 아이디와 이름 출력

← → ↻ ⓘ localhost:8080/jwbook/ch03/member/member2.jsp

아이디: 2022101

이름: 장그래

```
<jsp:useBean id="member" class="bean.MemberBean" scope="request"/>
```

```
<jsp:setProperty property="id" name="member"/>
```

```
<jsp:setProperty property="name" name="member"/>
```

생략 가능

```
<p>아이디 : <jsp:getProperty property="id" name="member"/>
```

```
<p>이름 : <jsp:getProperty property="name" name="member"/>
```



자바빈즈 액션 태그

setProperty 액션 태그로 아이디와 이름 설정하여 출력

← → ↻ ⓘ localhost:8080/jwbook/ch03/member/member3.jsp

아이디: 2022102

이름: 안영이

```
<jsp:useBean id="member" class="bean.MemberBean" scope="request"/>
<jsp:setProperty property="id" name="member" value="2021102"/>
<jsp:setProperty property="name" name="member" value="안영이"/>
```

```
<p>아이디 : <jsp:getProperty property="id" name="member"/>
<p>이름 : <jsp:getProperty property="name" name="member"/>
```



자바빈즈 액션 태그

Directive **page**로 **import** 하여 아이디와 이름 설정하여 출력

← → ↻ ⓘ localhost:8080/jwbook/ch03/member/member4.jsp

아이디: 2022101

이름: 장그래

```
<%@ page import="bean.MemberBean" %>
```

```
<body>
    <%
        MemberBean member = new MemberBean();

        /* out.println(member.getId());
        out.println(member.getName()); */
    %>
    <p>아이디 : <%=member.getId() %><br>
    <p>이름 : <%=member.getName() %>
</body>
```



JSP 빈즈 프로그래밍

사용자 로그인 구현

← → ↻ ⓘ localhost:8080/Chapter03/login/loginForm.jsp

아이디 :

비밀번호 :

← → ↻ ⓘ localhost:8080/Chapter03/login/loginProcess.jsp

로그인 성공!!

사용자 아이디 : myuser

사용자 패스워드 : 0000

프로그램 소스 목록

파일 이름	역 할
loginForm.html	사용자 로그인을 위해 아이디와 비밀번호 입력받는 화면
loginProcess.jsp	입력받은 아이디 정보를 빈즈 클래스를 이용 확인 처리한 화면
LoginBean.java	사용자가 입력한 계정 정보를 매핑하는 빈즈 클래스로써, 미리 저장된 계정 값과 비교해 로그인 성공 여부를 반환하는 메서드를 포함



JSP 빈즈 프로그래밍

사용자 로그인 구현

loginBean.jsp

```
package bean;

public class LoginBean {
    private String userid;
    private String passwd;

    final String _userid = "myuser";
    final String _passwd = "0000";

    //로그인 체크
    public boolean checkUser() {
        if(userid.equals(_userid) && passwd.equals(_passwd)) {
            return true;
        }else {
            return false;
        }
    }
}
```

```
public String getUserid() {
    return userid;
}

public void setUserid(String userid) {
    this.userid = userid;
}

public String getPasswd() {
    return passwd;
}

public void setPasswd(String passwd) {
    this.passwd = passwd;
}
```



JSP 빈즈 프로그래밍

사용자 로그인 구현

loginForm.jsp

```
<body>
  <form action="loginProcess.jsp" method="post">
    <p>
      <label for="userid">아이디 : </label>
      <input type="text" id="userid" name="userid">
    </p>
    <p>
      <label for="passwd">비밀번호 : </label>
      <input type="password" id="passwd" name="passwd">
    </p>
    <p><input type="submit" value="전송"></p>
  </form>
</body>
```



JSP 빈즈 프로그래밍

사용자 로그인 구현

loginProcess.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<jsp:useBean id="login" class="bean.LoginBean" scope="request" />
<jsp:setProperty property="userid" name="login"/>
<jsp:setProperty property="passwd" name="login"/>

<%
    if(login.checkUser())
        out.println("로그인 성공!!");
    else
        out.println("로그인 실패!!");
%>
<hr>
<p>사용자 아이디: <jsp:getProperty property="userid" name="login"/>
<p>사용자 패스워드: <jsp:getProperty property="passwd" name="login"/>
```



JSP 빈즈 프로그래밍

page import를 이용해 LoginBean 사용하기

```
<!-- page import로 LoginBean 가져와서 구현 -->
```

```
<%
```

```
    LoginBean login = new LoginBean();
```

loginProcess2.jsp

```
    String uid = request.getParameter("userid");
```

```
    String pwd = request.getParameter("passwd");
```

```
    login.setUserId(uid);
```

```
    login.setPasswd(pwd);
```

```
    if(login.checkUser()==true){  
        out.println("로그인 성공!!");
```

```
    }
```

```
    else{  
        out.println("로그인 실패!!");
```

```
    }
```

```
%>
```

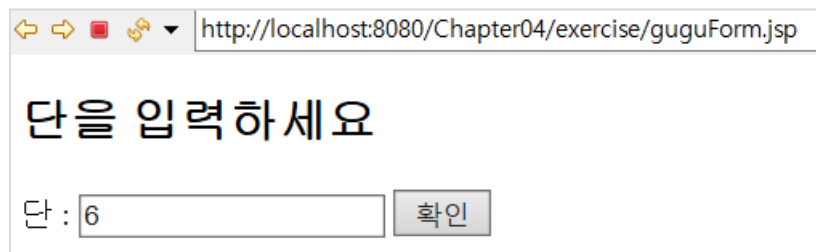
```
<p>아이디 : <%=login.getUserId() %>
```

```
<p>패스워드 : <%=login.getPasswd() %>
```

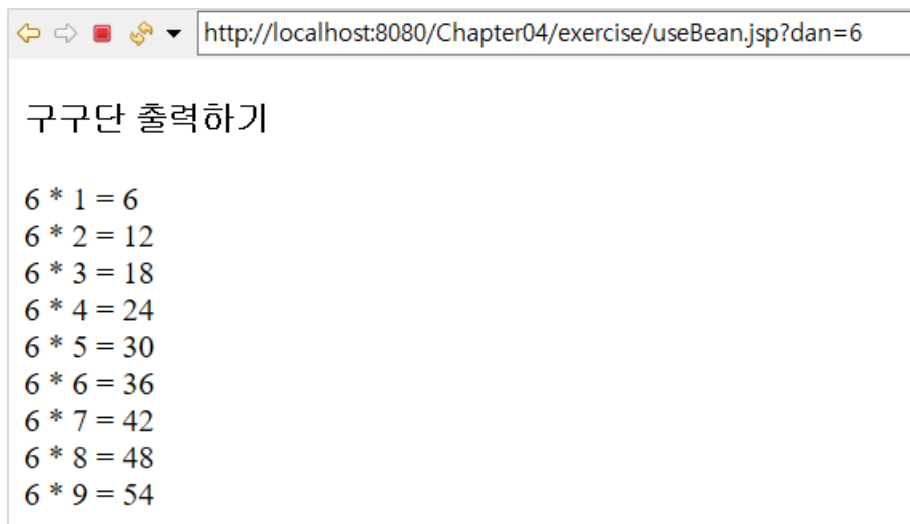


자바빈즈 액션 태그

구구단 애플리케이션 만들기



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/Chapter04/exercise/guguForm.jsp`. The page content displays the text "단을 입력하세요" (Please enter the multiplier). Below this text, there is a label "단 : " followed by a text input field containing the number "6". To the right of the input field is a button labeled "확인" (Confirm).



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/Chapter04/exercise/useBean.jsp?dan=6`. The page content displays the text "구구단 출력하기" (Print multiplication table). Below this text, the multiplication table for the number 6 is displayed, listing the products from 6 * 1 to 6 * 9.

6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54

자바빈즈 액션 태그

구구단 애플리케이션 만들기

guguForm.jsp

```
<h2>단을 입력하세요</h2>
<form action="useBean.jsp" method="get">
  <p>
    <label for="dan">단 : </label>
    <input type="text" name="dan">
    <input type="submit" value="확인">
  </p>
</form>
```

자바빈즈 액션 태그

Gugudan.java

```
public class Gugudan {  
    //구구단 곱셈 결과 메서드  
    public int[] process(int dan) {  
        int[] gugu = new int[9];  
        for(int i=1; i<10; i++) {  
            gugu[i-1] = dan * i;  
        }  
        return gugu;  
    }  
}
```

자바빈즈 액션 태그

useBean.jsp

```
<h4>구구단 출력하기</h4>
<jsp:useBean id="gugudan" class="ch04.com.dao.Gugudan" />
<%
    int dan = Integer.parseInt(request.getParameter("dan"));

    int[] result = gugudan.process(dan);
    for(int i=1; i<10; i++){
        out.println(dan + " * " + i + " = " + result[i-1] + "<br>");
    }
%>
```



엔터프라이즈 자바 빈즈(beans)

엔터프라이즈 자바 빈즈(Enterprise Java Beans, 이하 EJB)는 애플리케이션에서 비즈니스 로직을 구현하려는 컴포넌트 모델로, Java EE(Java Enterprise Edition)의 핵심이다.

Java EE는 보안을 중시하고 규모가 크며 확장 및 다른 시스템과의 상호, 운용을 필요로 하는 애플리케이션 개발에 필요한 핵심 기술이다.

상당수의 금융기관, 증권사, 이동통신사 등의 업무 시스템들이 Java EE를 기반으로 개발.

그림을 보면 엔터프라이즈 자바 빈즈는 데이터베이스 등 비즈니스 로직을 구현하게 되고 **JSP는 커다란 Java EE 스펙의 한 부분으로 웹 계층에** 위치하며 '프레젠테이션 레이어'라고 하는 사용자와의 상호작용을 담당하는 영역에 있다. 따라서 Java EE라를 큰 그림을 이해하고, 그 안에서 JSP 역할을 이해하는 노력을 기울여야 할 것이다.



엔터프라이즈 자바 빈즈(beans)

