

JSP 웹 프로그래밍

4 액션 태그

Contents

1

액션 태그의 개요

2

forward 액션 태그의 기능과 사용법

3

include 액션 태그의 기능과 사용법

4

Param 액션 태그의 기능과 사용법

5

자바빈즈 액션 태그의 기능과 사용법

6

[웹 쇼핑몰] 상품 목록 표시하기

학습목표

- 액션 태그의 개념과 특징을 이해합니다.
- 액션 태그 구성 요소의 사용법을 익힙니다.
- 액션 태그를 이용하여 웹 쇼핑몰의 상품 목록을 출력합니다.

1. 액션 태그의 개요

❖ 액션 태그(Action Tag)

- 서버나 클라이언트에게 어떤 행동을 하도록 명령하는 태그
- 주석, 디렉티브 태그와 함께 JSP 페이지를 구성하는 태그
- JSP 페이지에서 페이지와 페이지 사이를 제어하거나,
- 다른 페이지의 실행 결과 내용을 현재 페이지에 포함하거나,
- 자바 빈즈(JavaBeans) 등의 다양한 기능을 제공
- 액션 태그에는 다양한 종류가 있으며, `<%...%>`와 같은 스크립트 태그의 형식을 따르지 않고 XML 형식 `<jsp: ... />`를 사용
- 액션 태그는 반드시 끝나는 태그 `/>`로 마무리 해야 함.

1. 액션 태그의 개요

❖ 액션 태그의 종류

액션 태그	형식	설명
forward	<code><jsp:forward ... /></code>	다른 페이지로의 이동과 같은 페이지 흐름을 제어합니다.
include	<code><jsp:include ... /></code>	외부 페이지의 내용을 포함하거나 페이지를 모듈화합니다.
useBean	<code><jsp:useBean ... /></code>	JSP 페이지에 자바빈즈를 설정합니다.
setProperty	<code><jsp:setProperty ... /></code>	자바빈즈의 프로퍼티 값을 설정합니다.
getProperty	<code><jsp:getProperty ... /></code>	자바빈즈의 프로퍼티 값을 얻어옵니다.
param	<code><jsp:param ... /></code>	<code><jsp:forward></code> , <code><jsp:include></code> , <code><jsp:plugin></code> 태그에 인자를 추가합니다.
plugin	<code><jsp:plugin ... /></code>	웹 브라우저에 자바 애플릿을 실행합니다. 자바 플러그인에 대한 OBJECT 또는 EMBED 태그를 만드는 브라우저별 코드를 생성합니다.
element	<code><jsp:element ... /></code>	동적 XML 요소를 설정합니다.
attribute	<code><jsp:attribute ... /></code>	동적으로 정의된 XML 요소의 속성을 설정합니다.
body	<code><jsp:body ... /></code>	동적으로 정의된 XML 요소의 몸체를 설정합니다.
text	<code><jsp:text ... /></code>	JSP 페이지 및 문서에서 템플릿 텍스트를 작성합니다.

2. forward 액션 태그의 기능과 사용법

❖ forward 액션 태그

- 현재 JSP 페이지에서 다른 페이지로 이동하는 태그
- JSP 컨테이너는 현재 JSP 페이지에서 forward 액션 태그를 만나면
 - 그 전까지 출력 버퍼에 저장되어 있던 내용을 모두 삭제하고
 - forward 액션 태그에 설정된 페이지로 프로그램의 제어가 이동

`<jsp:forward page="파일명" />`

또는

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

반드시 끝나는 태그가 있어야 함

■ page 속성 값

- 현재 JSP 페이지에서 이동할 페이지의 외부 파일명
- 외부 파일은 현재 JSP 페이지와 같은 디렉터리에 있으면 파일명만 설정하고, 그렇지 않으면 전체 URL(또는 상대 경로)을 설정해야 함.

2. forward 액션 태그의 기능과 사용법

[forward 액션 태그 사용 예]

//forward.jsp

```
<% page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
  <h3>이 파일은 first.jsp입니다.</h3>
  <jsp:forward page="second.jsp" />

  <p>----- forward.jsp 페이지-----</p>
</body>
</html>
```

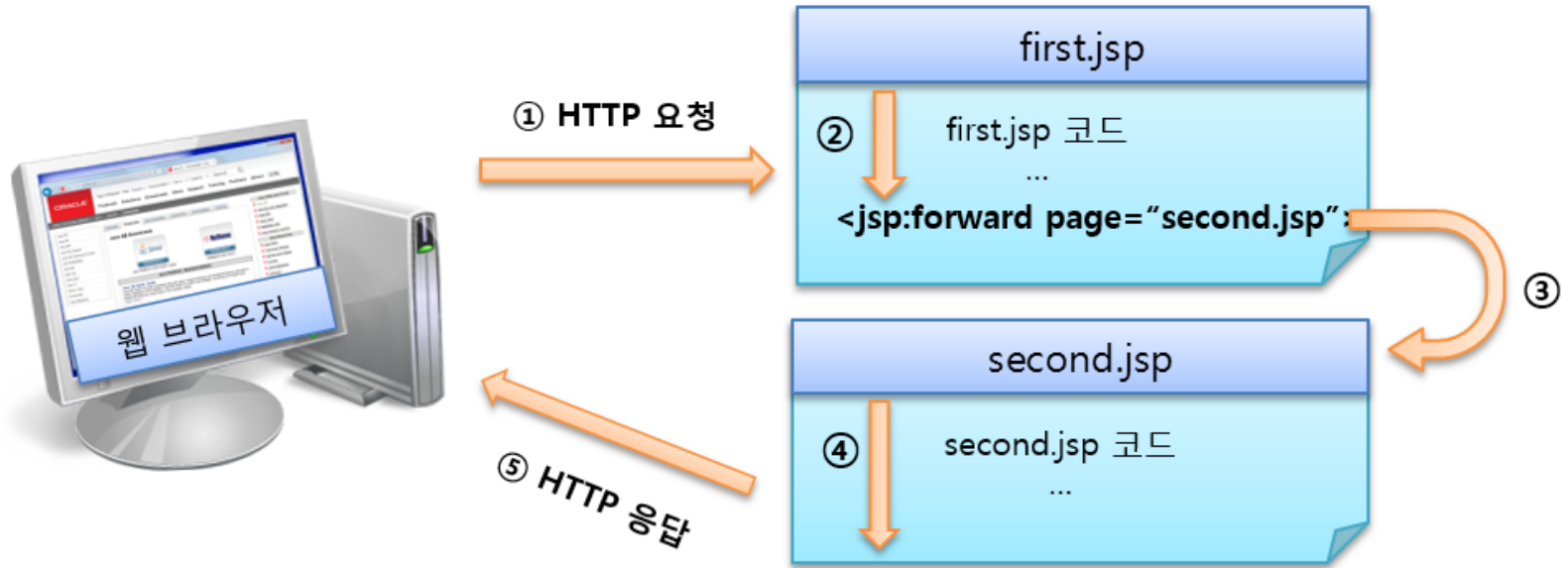
이 파일은 forward_date.jsp 입니다.

Today is : Wed Sep 05 15:35:07 KST 2018

```
//second.jsp
<% page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
  <h1> 이 파일은 forward_date.jsp 파일 입니다.
  <p>오늘의 날짜 및 시각
  <p><%= (new java.util.Date()).toLocaleString()%>
  <p><%= (new java.util.Date()) %>
  </h1>
</body>
</html>
```

2. forward 액션 태그의 기능과 사용법

❖ forward 액션 태그의 페이지 흐름 처리 과정



- ① 웹 브라우저에서 웹 서버로 first.jsp 를 요청함.
- ② JSP 컨테이너는 요청된 first.jsp 를 실행함.
- ③ first.jsp 를 실행하다가 forwara 액션 태그를 만나면 지금까지 저장된 출력 버퍼의 내용을 삭제하고 프로그램 제어를 page 속성에서 설정한 second.jsp 로 이동함(포워딩)
- ④ second.jsp 를 실행함.
- ⑤ JSP 컨테이너는 second.jsp 를 실행한 결과를 웹 브라우저에 응답으로 보냄.

2. forward 액션 태그의 기능과 사용법

예제 4-1 forward 액션 태그로 현재 날짜와 시각을 출력하는 페이지로 이동하기

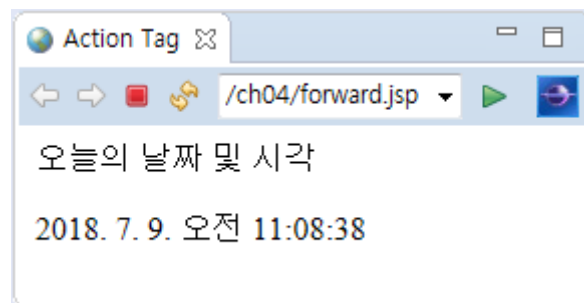
JSPBook/WebContent/ch04/forward.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <h2>forward 액션 태그</h2>
08     <jsp:forward page="forward_date.jsp" />
09     <p>_____
10 </body>
11 </html>
```


2. forward 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/forward_date.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <p>오늘의 날짜 및 시간
08     <p><%= (new java.util.Date()).toLocaleString() %>
09 </body>
10 </html>
```



3. include 액션 태그의 기능과 사용법

❖ include 액션 태그(1)

- include 디렉티브 태그처럼 현재 JSP 페이지의 특정 영역에 외부 파일의 내용을 포함하는 태그
- 현재 JSP 페이지에 포함할 수 있는 외부 파일은 HTML, JSP, 서블릿 페이지 등

```
<jsp:include page="파일명" flush="false"/>
```

■ page 속성 값

- 현재 JSP 페이지 내에 포함할 내용을 가진 외부 파일명
- 외부 파일은 현재 JSP 페이지와 같은 디렉터리에 있으면 파일명만 설정하고, 그렇지 않으면 전체 URL(또는 상대 경로)을 설정해야 함

3. include 액션 태그의 기능과 사용법

❖ include 액션 태그(2)

```
<jsp:include page="파일명" flush="false"/>
```

■ flush 속성 값

- 설정한 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장한 결과를 처리
- 기본 값은 false이며 출력 버퍼에 저장한 내용을 그대로 둠.
- true 로 설정하면 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장된 내용을 웹 브라우저에 출력하고 출력 버퍼를 비움.
 - 일단 출력버퍼의 내용을 웹 브라우저에 전송하게 되는데 이때 헤더정보도 같이 전송. 헤더정보가 일단 웹 브라우저에 전송이 되고 나면 헤더정보를 추가해도 결과가 반영되지 않음.

3. include 액션 태그의 기능과 사용법

❖ Include디렉티브 태그와 Include액션 태그의 차이점

- 두 태그의 차이점
 - Include액션 태그에서는 포함된 외부 파일이 실행된 후 현재 JSP 페이지로 제어를 반환한다는 것.
- include 디렉티브는 단순히 소스의 내용이 텍스트로 포함되지만 <jsp:include> 액션 태그는 포함시킬 페이지의 처리 결과를 포함시킨다는 점이 다름. 포함되는 페이지는 html, jsp, Servlet페이지 모두 가능.
- include 디렉티브는 주로 조각코드를 삽입할 때 사용되고, <jsp:include> 액션 태그는 페이지를 모듈화 할 때 사용됨. 즉, 템플릿 페이지를 작성할 때 사용.

3. include 액션 태그의 기능과 사용법

[include 액션 태그 사용 예]

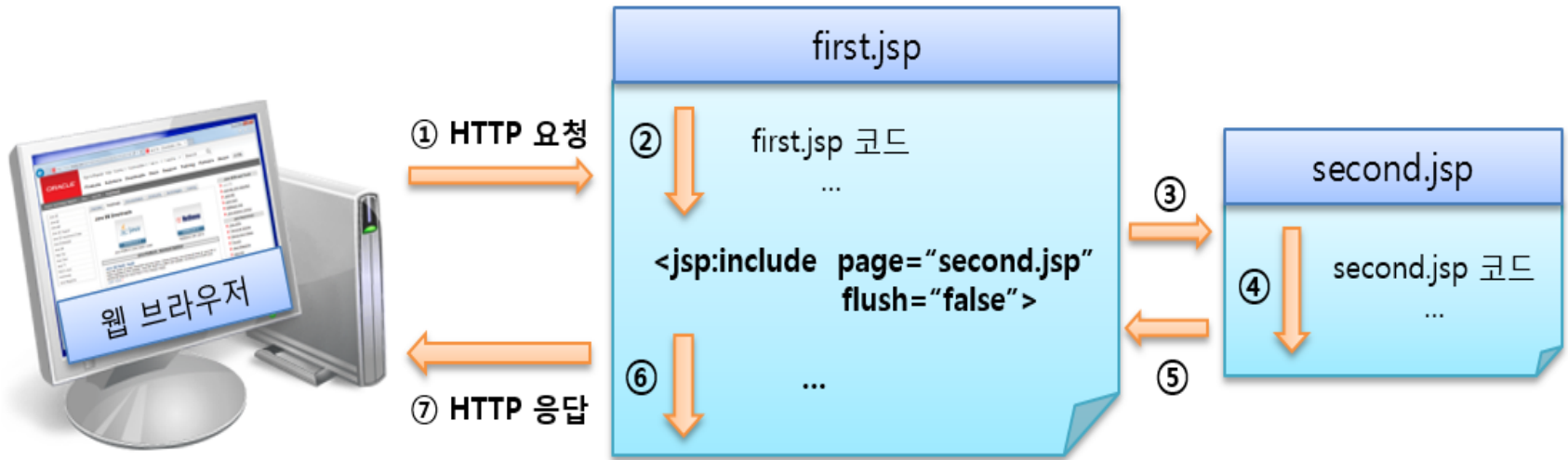
```
//first.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
  <h3>이 파일은 first.jsp입니다.</h3>
  <jsp:include page="second.jsp" flush="false" />
  <p>Java Server Page</p>
</body>
</html>
```

```
//second.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
  <h3>이 파일은 second.jsp입니다.</h3>
  Today is : <%=new java.util.Date()%>
</body>
</html>
```

이 파일은 **first.jsp**입니다.
이 파일은 **second.jsp**입니다.
Today is : Wed Sep 05 15:38:45 KST 2018
Java Server Page

3. include 액션 태그의 기능과 사용법

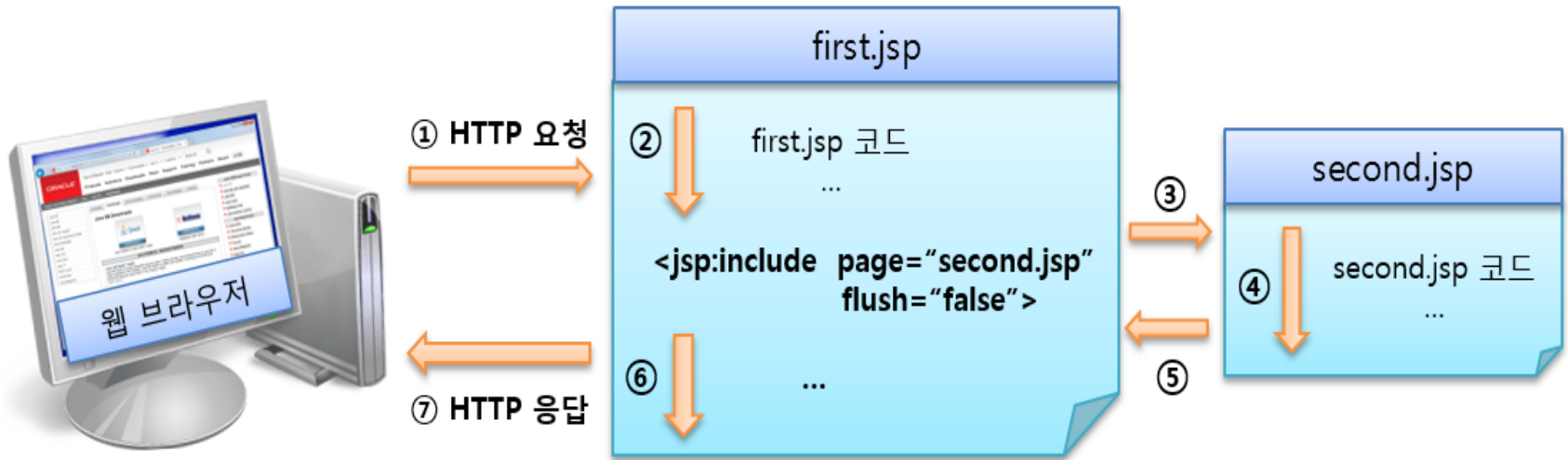
❖ include 액션 태그의 처리 과정(1)



- ① 웹 브라우저에서 웹 서버로 first.jsp 를 요청함.
- ② JSP 컨테이너는 요청받은 first.jsp 를 처리하고 first.jsp 내의 출력 내용이 출력 버퍼에 저장됨.
- ③ 이때, `<jsp :include page="second.jsp" flush="false" />`문장을 만나면 하던 작업을 멈추고 프로그램 제어를 second.jsp 로 이동함
- ④ second.jsp 를 실행하고 second.jsp 내의 출력 내용이 출력 버퍼에 저장됨.

3. include 액션 태그의 기능과 사용법

❖ include 액션 태그의 처리 과정



- ⑤ `second.jsp`의 처리가 끝나면 다시 `first.jsp` 로 프로그램의 제어가 이동하는데, 이동 위치는 `<jsp:include page="second.jsp" flush="false" />` 문장의 다음 행이 됨.
- ⑥ `first.jsp` 의 나머지 부분을 처리하고, 출력할 내용이 있으면 출력 버퍼에 저장함.
- ⑦ JSP 컨테이너는 출력 버퍼의 내용을 웹 브라우저에 응답으로 보냄.

3. include 액션 태그의 기능과 사용법

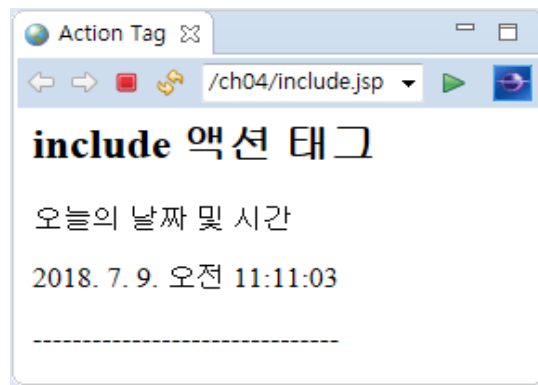
구분	include 액션 태그	include 디렉티브 태그
처리 시간	요청 시 자원을 포함합니다.	번역 시 자원을 포함합니다.
기능	별도의 파일로 요청 처리 흐름을 이동합니다.	현재 페이지에 삽입합니다.
데이터 전달 방법	request 기본 내장 객체나 param 액션 태그를 이용하여 파라미터를 전달합니다.	페이지 내의 변수를 선언한 후 변수에 값을 저장합니다.
용도	화면 레이아웃의 일부분을 모듈화할 때 주로 사용합니다.	다수의 JSP 웹 페이지에서 공통으로 사용되는 코드나 저작권과 같은 문장을 포함하는 경우에 사용합니다.
기타	동적 페이지에 사용합니다.	정적 페이지에 사용합니다.

3. include 액션 태그의 기능과 사용법

예제 4-2 include 액션 태그에 현재 날짜와 시각을 출력하는 페이지 포함하기

JSPBook/WebContent/ch04/include.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <h2>include 액션 태그</h2>
08     <jsp:include page="include_date.jsp" flush="true" />
09     <p>-----
10 </body>
11 </html>
```



3. include 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/include_date.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <p>오늘의 날짜 및 시각</p>
08     <p>X%=(new java.util.Date()).toLocaleString()%X/p>
09 </body>
10 </html>
```

4. param 액션 태그의 기능과 사용법

❖ param 액션 태그

- 현재 JSP 페이지에서 다른 페이지에 정보를 전달하는 태그
- 이 태그는 단독으로 사용되지 못하며 <jsp:forward>나 <jsp:include> 태그의 내부에 사용
- 다른 페이지에 여러 개의 정보를 전송해야 할 때는 다중의 param 액션 태그를 사용

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

4. param 액션 태그의 기능과 사용법

[jsp:param 액션 태그 사용 예]

```
//first.jsp
<% page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
    <h3>이 파일은 first.jsp입니다.</h3>
    <jsp:include page="second.jsp">
        <jsp:param name="date" value="<%=new java.util.Date()%%" />

    </jsp:include>
    <p>Java Server Page</p>
</body>
</html>
```

이 파일은 **first.jsp**입니다.

Today is : Wed Sep 05 15:43:02 KST 2018

Java Server Page

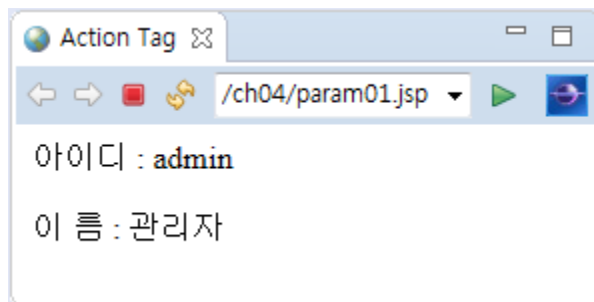
```
//second.jsp
<% page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Action Tag</title>
</head>
<body>
    Today is : <%=request.getParameter("date")%>
</body>
</html>
```

4. param 액션 태그의 기능과 사용법

예제 4-3 forward 액션 태그와 param 액션 태그에 아이디와 이름 전달하기

JSPBook/WebContent/ch04/param01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <h3>param 액션 태그</h3>
08     <jsp:forward page="param01_data.jsp">
09         <jsp:param name="id" value="admin" />
10         <jsp:param name="name" value='<%=java.net.URLEncoder.encode("관리자")%>' />
11     </jsp:forward>
12     <p>Java Server Page
13 </body>
14 </html>
```



4. param 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/param01_data.jsp

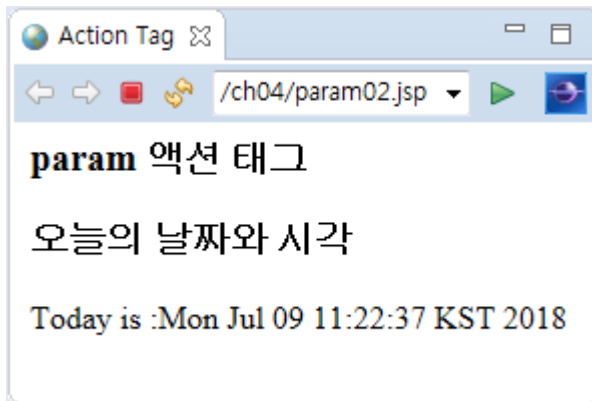
```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <p> 아이디 : <%=request.getParameter("id")%>
08         <%
09             String name = request.getParameter("name");
10             %>
11     <p> 이 름 : <%=java.net.URLDecoder.decode(name)%>
12 </body>
13 </html>
```

4. param 액션 태그의 기능과 사용법

예제 4-4 include 액션 태그와 param 액션 태그에 제목과 현재 날짜 전달하기

JSPBook/WebContent/ch04/param02.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <h3>param 액션 태그</h3>
08     <jsp:include page="param02_data.jsp">
09         <jsp:param name="title" value='<%=java.
10             net.URLEncoder.encode("오늘의 날짜와 시각")%>' />
11         <jsp:param name="date" value="<%=java.util.Calendar.getInstance().
12             getTime()%>" />
13     </jsp:include>
14 </body>
15 </html>
```



4. param 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/param02_data.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <%
08         String title = request.getParameter("title");
09     %>
10     <h3><%=java.net.URLDecoder.decode(title)%></h3>
11     Today is :<%=request.getParameter("date")%>
12 </body>
13 </html>
```


5. 자바빈즈 액션 태그의 기능과 사용법

❖ 자바빈즈(Java Beans)

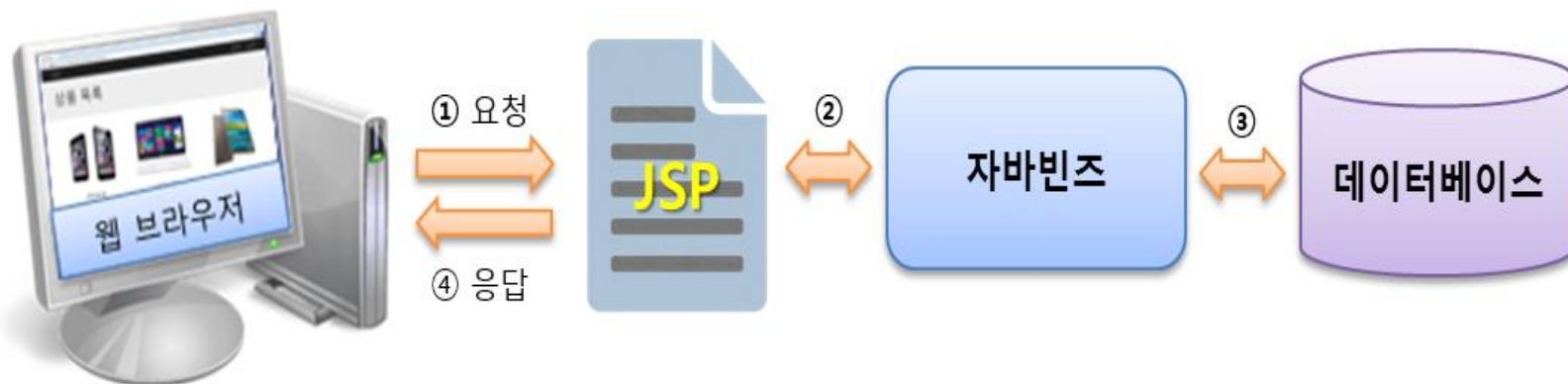
- 프로그램의 효율을 높이기 위해 화면을 출력하는 부분과 데이터를 처리하는 로직 부분을 작성하며, **로직 부분의 코드에 자바빈즈(Java Beans)라는 클래스를 사용함.**
- 동적 콘텐츠 개발을 위해 자바 코드를 사용하여 자바 클래스로 로직을 작성하는 방법
- **JSP 페이지에서 화면을 표현하기 위한 계산식이나 자료의 처리를 담당하는 자바코드를 따로 분리하여 작성하는 것**
- JSP 페이지가 HTML과 같이 쉽고 간단한 코드만으로 구성

5. 자바빈즈 액션 태그의 기능과 사용법

❖ 자바빈즈(Java Beans)

■ JSP와 자바빈즈의 흐름

- ① 웹 브라우저가 JSP 페이지에 요청을 전송함.
- ② JSP 페이지는 자바빈즈와 통신함.
- ③ 자바빈즈가 데이터베이스에 연결됨
- ④ JSP 페이지가 브라우저에 응답함.



5. 자바빈즈 액션 태그의 기능과 사용법

- ❖ 자바빈즈는 데이터 표현을 목적으로 하는 자바 클래스이므로 기존의 자바 클래스를 작성하는 방법과 동일하게 작성함.
- ❖ 자바빈즈는 데이터를 담는 **멤버 변수인 프로퍼티(property)**와 데이터를 가져오거나 저장하는 **메서드**로 구성됨.
- ❖ 자바빈즈를 작성할 때는 다음 규칙을 따라야 합니다.
 - ❶ 자바 클래스는 `java.io.Serializable` 인터페이스를 구현해야 합니다.
 - ❷ 인수가 없는 기본 생성자가 있어야 합니다.
 - ❸ 모든 멤버 변수인 프로퍼티는 `private` 접근 지정자로 설정해야 합니다.
 - ❹ 모든 멤버 변수인 프로퍼티는 `getter/setter()` 메소드가 존재해야 합니다.
 - `setter()` 메소드는 멤버 변수에 값을 저장할 수 있는 메소드입니다.
 - `getter()` 메소드는 멤버 변수에 저장된 값을 가져올 수 있는 메소드입니다.

5. 자바빈즈 액션 태그의 기능과 사용법

[자바빈즈 작성 예]

```
package com.dao;
import java.io.Serializable;

public class MemberBean implements java.io.Serializable {
    private int id;
    private String name; } ③ 멤버 변수(프로퍼티) 정의
    public MemberBean(){ } ② 기본 생성자
    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;
    }
}
```

①

④ getter/setter() 메소드

5. 자바빈즈 액션 태그의 기능과 사용법

❖ useBean 액션 태그

- JSP 페이지에서 자바빈즈를 사용하기 위해 실제 자바 클래스를 선언하고 초기화하는 태그
- id 속성과 scope 속성을 바탕으로 자바빈즈의 객체를 검색하고, 객체가 발견되지 않으면 빈 객체를 생성

객체 이름

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

속성	설명
id	자바빈즈를 식별하기 위한 이름입니다.
class	패키지 이름을 포함한 자바빈즈 이름입니다. 자바빈즈는 인수가 없는 기존 생성자가 있어야 하며 추상 클래스를 사용할 수 없습니다.
scope	자바빈즈가 저장되는 영역을 설정합니다. page(기본 값), request, session, application 중 하나의 값을 사용합니다.

5. 자바빈즈 액션 태그의 기능과 사용법

[useBean 액션 태그 사용 예]

```
<jsp:useBean id="member" class="com.dto.MemberBean" scope="page" />
```

|| (아래 Java Code와 동일함)

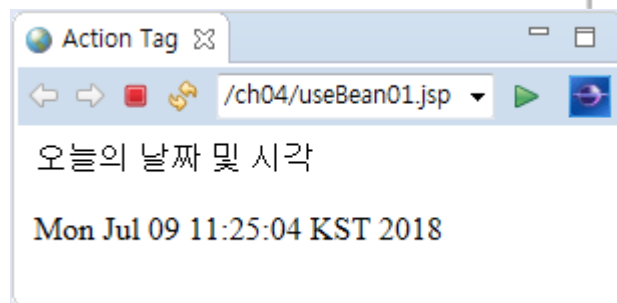
```
MemberBean member = (MemberBean) request.getAttribute ( "member");  
if (member == null){  
    member = new MemberBean();  
    request.setAttribute("member", member);  
}
```

5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-5 useBean 액션 태그에 Date 클래스를 사용하여 현재 날짜와 시각 출력하기

JSPBook/WebContent/ch04/useBean01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="date" class="java.util.Date" />
08     <p> <%
09         out.print("오늘의 날짜 및 시각");
10     %>
11     <p><%=date%>
12 </body>
13 </html>
```



5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-6 자바빈즈 Calculator를 생성하고 useBean 액션 태그에 Calculator 클래스를 사용하여 숫자 출력하기

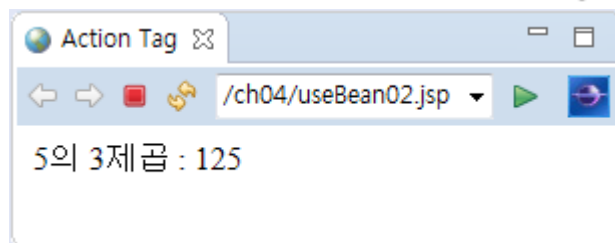
JSPBook/src/ch04/com/dao/Calculator.java

```
01 package ch04.com.dao;  
02  
03 public class Calculator {  
04     public int process(int n) {  
05         return n * n * n;  
06     }  
07 }
```


5. 자바빈즈 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/useBean02.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="bean" class="ch04.com.dao.Calculator" />
08     <%
09         int m = bean.process(5);
10         out.print("5의 3제곱 : " + m);
11     %>
12 </body>
13 </html>
```



5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-7 자바빈즈 Person을 생성하고 useBean 액션 태그에 Person 클래스를 사용하여 아이디와 이름 출력하기

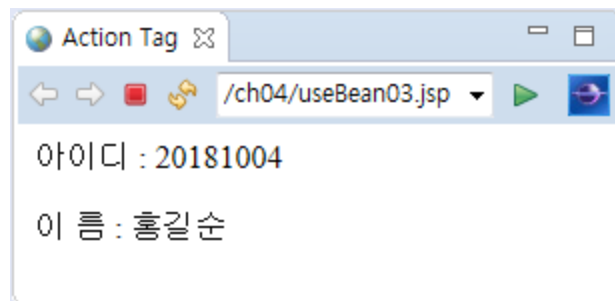
JSPBook/src/ch04/com/dao/Person.java

```
01 package ch04.com.dao;
02
03 public class Person {
04     private int id = 20181004;
05     private String name = "홍길순";
06
07     public Person() {
08
09     }
10
11     public int getId() {
12         return id;
13     }
14
15     public void setId(int id) {
16         this.id = id;
17     }
18
19     public String getName() {
20         return name;
21     }
22
23     public void setName(String name) {
24         this.name = name;
25     }
26 }
```

5. 자바빈즈 액션 태그의 기능과 사용법

JSPBook/WebContent/ch04/useBean03.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="person" class="ch04.com.dao.Person" scope="request" />
08     <p> 아이디 : <%=person.getId()%>
09     <p> 이 름 : <%=person.getName()%>
10 </body>
11 </html>
```

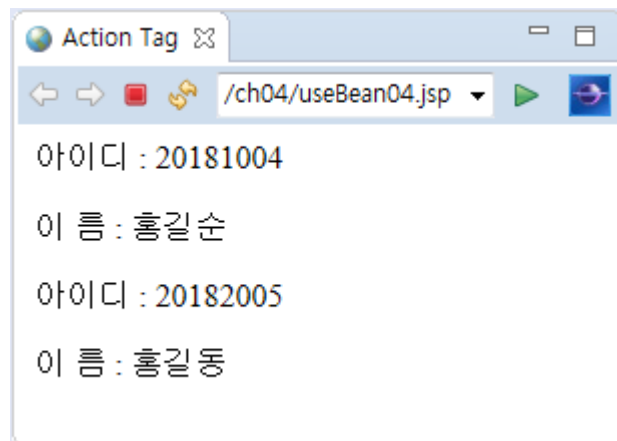


5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-8 useBean 액션 태그에 [예제 4-7]에서 생성한 자바빈즈 Person으로 아이디와 이름을 설정하여 출력하기

JSPBook/WebContent/ch04/useBean04.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="person" class="ch04.com.dao.Person" scope="request" />
08     <p> 아이디 : <%=person.getId()%>
09     <p> 이 름 : <%=person.getName()%>
10         <%
11             person.setId(20182005);
12             person.setName("홍길동");
13         %>
14     <jsp:include page="useBean03.jsp">
15 </body>
16 </html>
```



5. 자바빈즈 액션 태그의 기능과 사용법

❖ setProperty 액션 태그

- 프로퍼티의 값 저장하기
- useBean 액션 태그와 함께 자바빈즈의 setter() 메소드에 접근하여 자바빈즈의 멤버 변수인 프로퍼티의 값을 저장하는 태그

`<jsp:setProperty name="자바빈즈 식별이름" property="프로퍼티 이름" value="값" />`

A blue line points from the text '객체 이름' (Object Name) in a blue box to the 'name' attribute in the JSP tag code snippet.

객체 이름

- 폼 페이지로부터 전달되는 요청 파라미터의 값을 직접 저장하거나 자바빈즈의 프로퍼티로 변경하여 값을 저장할 수 있습니다.
- 또는 모든 자바빈즈 프로퍼티 이름과 동일하게 요청 파라미터를 설정할 수 있습니다.

5. 자바빈즈 액션 태그의 기능과 사용법

[setProperty 액션 태그 사용 예]

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

속성	설명
name	useBean 태그에 id 속성 값으로 설정된 자바빈즈를 식별하기 위한 이름입니다.
property	자바빈즈의 프로퍼티 이름입니다. 만약 프로퍼티 이름에 '*'를 사용하면 모든 요청 파라미터가 자바빈즈 프로퍼티의 setter() 메소드에 전달됨을 의미합니다.
value	변경할 자바빈즈의 프로퍼티 값입니다. 만약 프로퍼티 값이 null이거나 존재하지 않는 요청 파라미터인 경우에는 setProperty 액션 태그가 무시됩니다.
param	자바빈즈의 프로퍼티 값을 전달하는 요청 파라미터의 이름입니다. param과 value를 동시에 모두 사용할 수 없으며 하나를 선택하여 사용하는 것은 가능합니다.

5. 자바빈즈 액션 태그의 기능과 사용법

```
<jsp:useBean id="member" class="com.dto.MemberBean" scope="page" />
```

[자바빈즈의 프로퍼티 값 출력 예]

```
<% out.println("아이디 : "+member.getId()); %>
```

5. 자바빈즈 액션 태그의 기능과 사용법

- 요청 파라미터 이름과 자바빈즈의 프로퍼티 이름이 일치하는 경우:

```
//폼 페이지  
<form action="memberProcess.jsp" method="post">  
    <input name="id" value="admin" />  
</form>
```

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


5. 자바빈즈 액션 태그의 기능과 사용법

- 요청 파라미터 이름과 자바빈즈의 프로퍼티 이름이 일치하지 않는 경우:

//폼 페이지

```
<form action="memberProcess.jsp" method="post">  
    <input name="userId" value="admin" />  
</form>
```

//jsp 페이지

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

5. 자바빈즈 액션 태그의 기능과 사용법

- 요청 파라미터 이름과 자바빈즈의 프로퍼티 이름이 모두 일치하는 경우:

//폼 페이지

```
<form action="memberProcess.jsp" method ="post">
```

```
    <input name="id" value="admin" />
```

```
    <input name="name" value="관리자" />
```

```
</form>
```

//jsp 페이지

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

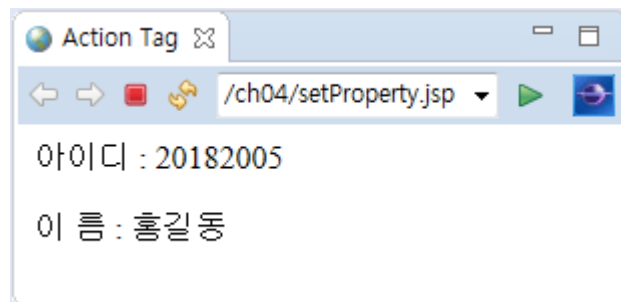
5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-9 setProperty 액션 태그에 자바빈즈 Person으로 아이디와 이름을 설정하여 출력하기

JSPBook/WebContent/ch04/setProperty.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>

03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="person" class="ch04.com.dao.Person" scope="request" />
08     <jsp:setProperty name="person" property="id" value="20182005" />
09     <jsp:setProperty name="person" property="name" value="홍길동" />
10     <p> 아이디 : <% out.println(person.getId()); %>
11     <p> 이 름 : <% out.println(person.getName()); %>
12 </body>
13 </html>
```



5. 자바빈즈 액션 태그의 기능과 사용법

❖ getProperty 액션 태그

- **프로퍼티의 값 가져오기**
- useBean 액션 태그와 함께 자바빈즈의 getter() 메소드에 접근하여 자바빈즈의 멤버 변수인 프로퍼티의 값을 가져오는 태그

`<jsp:getProperty name="자바빈즈 식별이름" property="프로퍼티 이름" />`

속성	설명
name	useBean 태그에 id 속성 값으로 설정된 자바빈즈를 식별하기 위한 이름입니다.
property	자바빈즈의 프로퍼티 이름입니다. 만약 프로퍼티 이름에 '*'를 사용하면 모든 요청 파라미터가 자바빈즈 프로퍼티의 getter() 메소드에 전달됨을 의미합니다.

5. 자바빈즈 액션 태그의 기능과 사용법

[getProperty 액션 태그 사용 예]

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

[자바빈즈의 프로퍼티 값 출력 예]

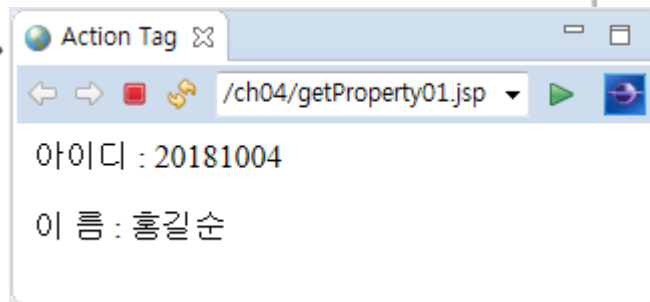
```
<% out.println(member.getName()); %>
```

5. 자바빈즈 액션 태그의 기능과 사용법

예제 4-10 getProperty 액션 태그에 자바빈즈 Person을 이용하여 아이디와 이름을 가져와 출력하기

JSPBook/WebContent/ch04/getProperty01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="person" class="ch04.com.dao.Person" scope="request" />
08     <p> 아이디 : <jsp:getProperty name="person" property="id" />
09     <p> 이 름 : <jsp:getProperty name="person" property="name" />
10 </body>
11 </html>
```

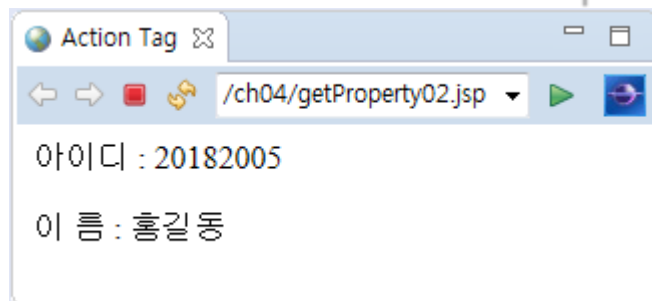


5. 자바빈즈 액션 태그의 기능과 사용법

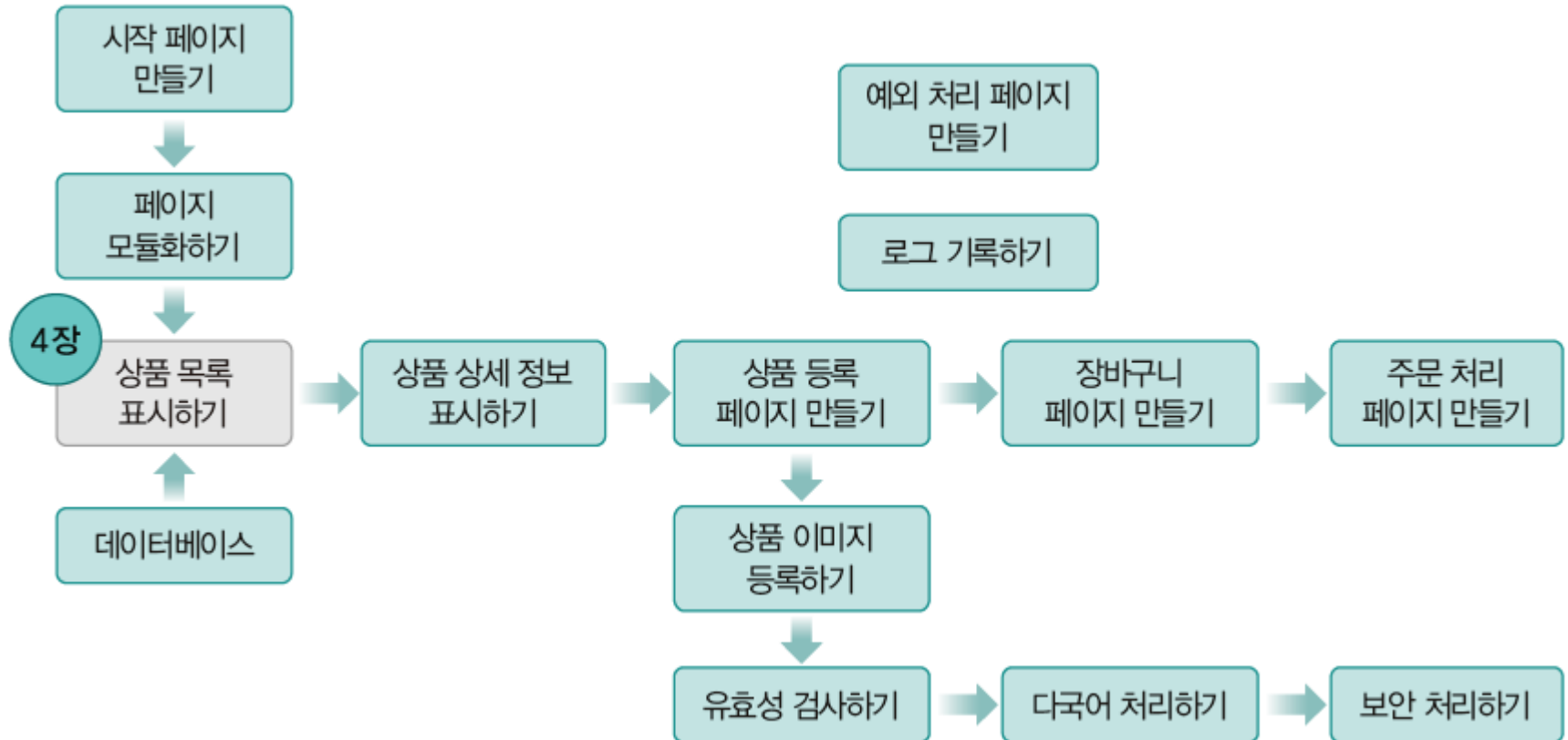
예제 4-11 getProperty 액션 태그에 자바빈즈 Person을 이용하여 폼 페이지에 입력한 아이디와 이름을 전달받아 출력하기

JSPBook/WebContent/ch04/getProperty02.jsp

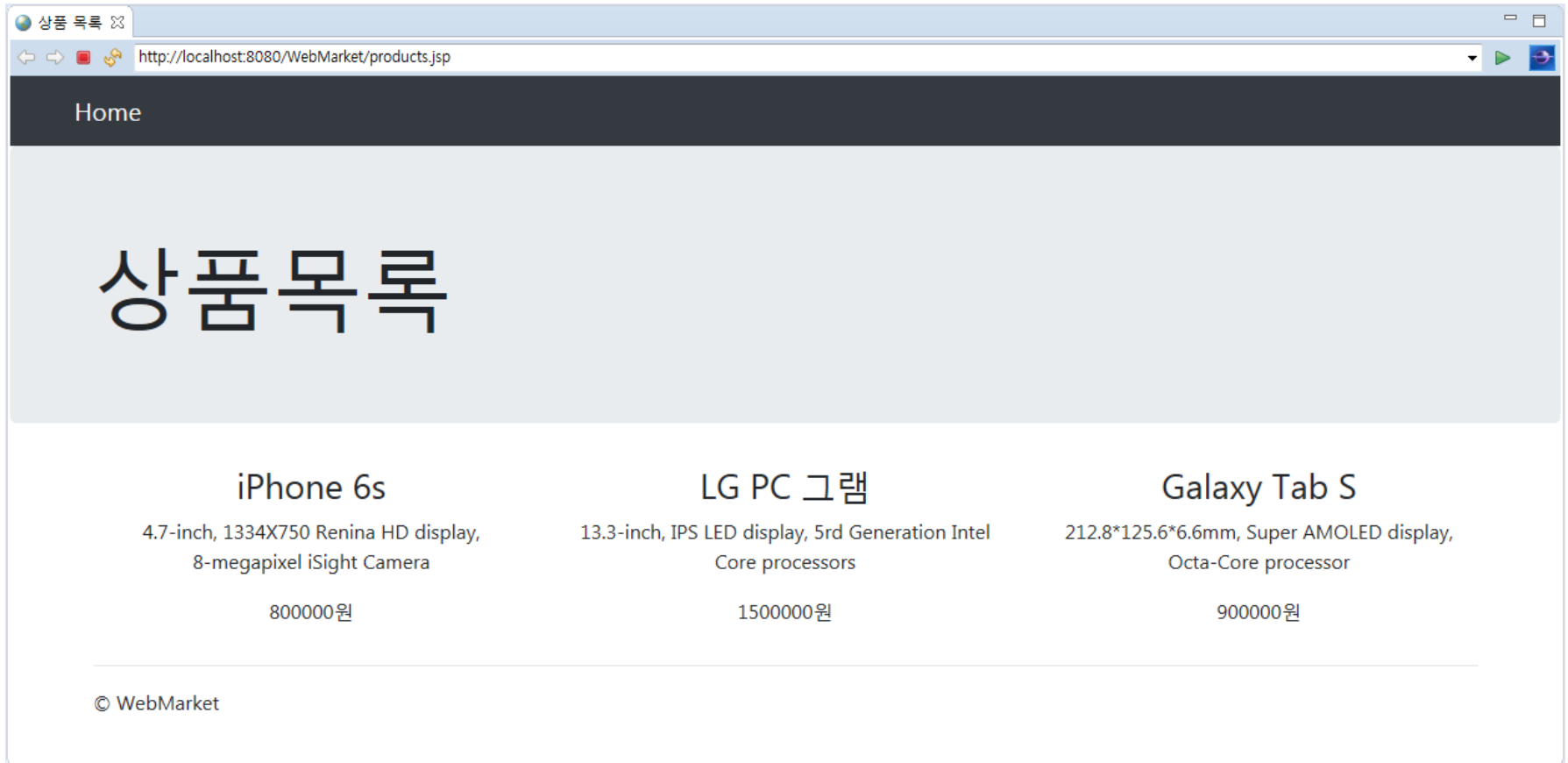
```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Action Tag</title>
05 </head>
06 <body>
07     <jsp:useBean id="person" class="ch04.com.dao.Person">/jsp:useBean>
08     <jsp:setProperty name="person" property="id" value="20182005" />
09     <jsp:setProperty name="person" property="name" value="홍길동" />
10     <p> 아이디 : <jsp:getProperty property="id" name="person" />
11     <p> 이 름 : <jsp:getProperty property="name" name="person" />
12 </body>
13 </html>
```



6. [웹 쇼핑몰] 상품 목록 표시하기



6. [웹 쇼핑몰] 상품 목록 표시하기



6. [웹 쇼핑몰] 상품 목록 표시하기

예제 4-12 상품 목록 표시하기

- 상품 클래스 생성하기:
- 멤버 변수 선언하기:

WebMarket/src/dto/Product.java

```
01 package dto;
02
03 import java.io.Serializable;
04
05 public class Product implements Serializable{
06
07     private static final long serialVersionUID = -4274700572038677000L;
08
09     private String productId;           //상품 아이디
10     private String pname;              //상품명
11     private Integer unitPrice;         //상품 가격
12     private String description;        //상품 설명
13     private String manufacturer ;      //제조사
14     private String category;           //분류
15     private long unitsInStock;         //재고 수
16     private String condition;          //신상품 or 중고품 or 재생품
17 }
```

6. [웹 쇼핑몰] 상품 목록 표시하기

■ 기본 생성자 작성하기

WebMarket/src/dto/Product.java

```
01 package dto;
02
03 import java.io.Serializable;
04
05 public class Product implements Serializable{
06     ...(생략)...
07     public Product() {
08         super();
09     }
10
11     public Product(String productId, String pname, Integer unitPrice) {
12         this.productId = productId;
13         this.pname = pname;
14         this.unitPrice = unitPrice;
15     }
16 }
```

6. [웹 쇼핑몰] 상품 목록 표시하기

- 모든 멤버 변수의 Setter/Getter() 메소드 작성하기:

```
01 package dto;
02
03 import java.io.Serializable;
04
05 public class Product implements Serializable{
06     ...(생략)...
07     public String getProductId() {
08         return productId;
09     }
10
11     public String getPname() {
12         return pname;
13     }
14
15     public void setPname(String pname) {
16         this.pname = pname;
17     }
18
19     public void setProductId(String productId) {
20         this.productId = productId;
21     }
22
```

WebMarket/src/dto/Product.java

```
22
23     public Integer getUnitPrice() {
24         return unitPrice;
25     }
26
27     public void setUnitPrice(Integer unitPrice) {
28         this.unitPrice = unitPrice;
29     }
30
31     public String getDescription() {
32         return description;
33     }
34
35     public void setDescription(String description) {
36         this.description = description;
37     }
38
39     public String getManufacturer() {
40         return manufacturer;
41     }
42
43     public void setManufacturer(String manufacturer) {
44         this.manufacturer = manufacturer;
45     }
46
47     public String getCategory() {
48         return category;
49     }
50
51     public void setCategory(String category) {
52         this.category = category;
53     }
54
55     public long getUnitsInStock() {
56         return unitsInStock;
57     }
58
59     public void setUnitsInStock(long unitsInStock) {
60         this.unitsInStock = unitsInStock;
61     }
62
63     public String getCondition() {
64         return condition;
65     }
66
67     public void setCondition(String condition) {
68         this.condition = condition;
69     }
70 }
```

6. [웹 쇼핑몰] 상품 목록 표시하기

❖ [자바빈즈로 사용할 상품 데이터 접근 클래스 만들기]

- 자바빈즈로 사용할 클래스 만들기:
- 멤버 변수와 기본 생성자 만들기:

WebMarket/src/dao/ProductRepository.java

```
01 package dao;
02
03 import java.util.ArrayList;
04
05 import dto.Product;
06
07 public class ProductRepository {
08
09     private ArrayList<Product> listOfProducts = new ArrayList<Product>();
```

6. [웹 쇼핑몰] 상품 목록 표시하기

```
10
11     public ProductRepository() {
12         Product phone = new Product("P1234", "iPhone 6s", 8000000);
13         phone.setDescription("4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera");
14         phone.setCategory("Smart Phone");
15         phone.setManufacturer("Apple");
16         phone.setUnitsInStock(1000);
17         phone.setCondition("New");
18
19         Product notebook = new Product("P1235", "LG PC 그램", 15000000);
20         notebook.setDescription("13.3-inch, IPS LED display, 5rd Generation Intel Core processors");
21         notebook.setCategory("Notebook");
22         notebook.setManufacturer("LG");
23         notebook.setUnitsInStock(1000);
24         notebook.setCondition("Refurbished");
25
26         Product tablet = new Product("P1236", "Galaxy Tab S", 9000000);
27         tablet.setDescription("212.8*125.6*6.6mm, Super AMOLED display, Octa-Core processor");
28         tablet.setCategory("Tablet");
29         tablet.setManufacturer("Samsung");
30         tablet.setUnitsInStock(1000);
31         tablet.setCondition("Old");
32
33         listOfProducts.add(phone);
34         listOfProducts.add(notebook);
35         listOfProducts.add(tablet);
36     }
37 }
```

6. [웹 쇼핑몰] 상품 목록 표시하기

- 상품 목록을 가져오는 메소드 만들기:

WebMarket/src/dao/ProductRepository.java

```
01 package dao;
02 ...(생략)...
03 public class ProductRepository {
04     ...(생략)...
05     public ArrayList<Product> getAllProducts() {
06         return listOfProducts;
07     }
08 }
```

6. [웹 쇼핑몰] 상품 목록 표시하기

❖ [상품 목록 표시하기]

- 상품 목록 출력 웹 페이지 만들기:

WebMarket/WebContent/products.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="java.util.ArrayList"%>
03 <%@ page import="dto.Product"%>
04 <jsp:useBean id="productDAO" class="dao.ProductRepository" scope="session" />
05 <html>
06 <head>
07 <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
08 <title>상품 목록</title>
09 </head>
10 <body>
11 <jsp:include page="menu.jsp" />
12 <div class="jumbotron">
13 <div class="container">
14 <h1 class="display-3">상품 목록</h1>
```


6. [웹 쇼핑몰] 상품 목록 표시하기

```
15         </div>
16     </div>
17     <%
18         ArrayList<Product> listOfProducts = productDAO.getAllProducts();
19     %>
20     <div class="container">
21         <div class="row" align="center">
22             <%
23                 for (int i = 0; i < listOfProducts.size(); i++) {
24                     Product product = listOfProducts.get(i);
25                 %>
26                 <div class="col-md-4">
27                     <h3>X%=product.getPname()%</h3>
28                     <p>X%=product.getDescription()%>
29                     <p>X%=product.getUnitPrice()%>원
30                 </div>
31             <%
32                 }
33             %>
34         </div>
35         <hr>
36     </div>
37     <jsp:include page="footer.jsp" />
38 </body>
39 </html>
```