

# 복습하기

## 웹정보시스템과 SQL의 DML

데이터 검색

• SELECT 문

데이터 입력

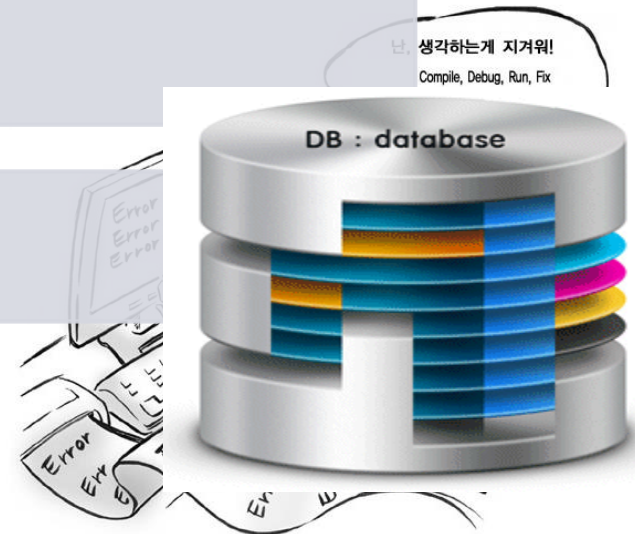
• INSERT 문

데이터 수정

• UPDATE 문

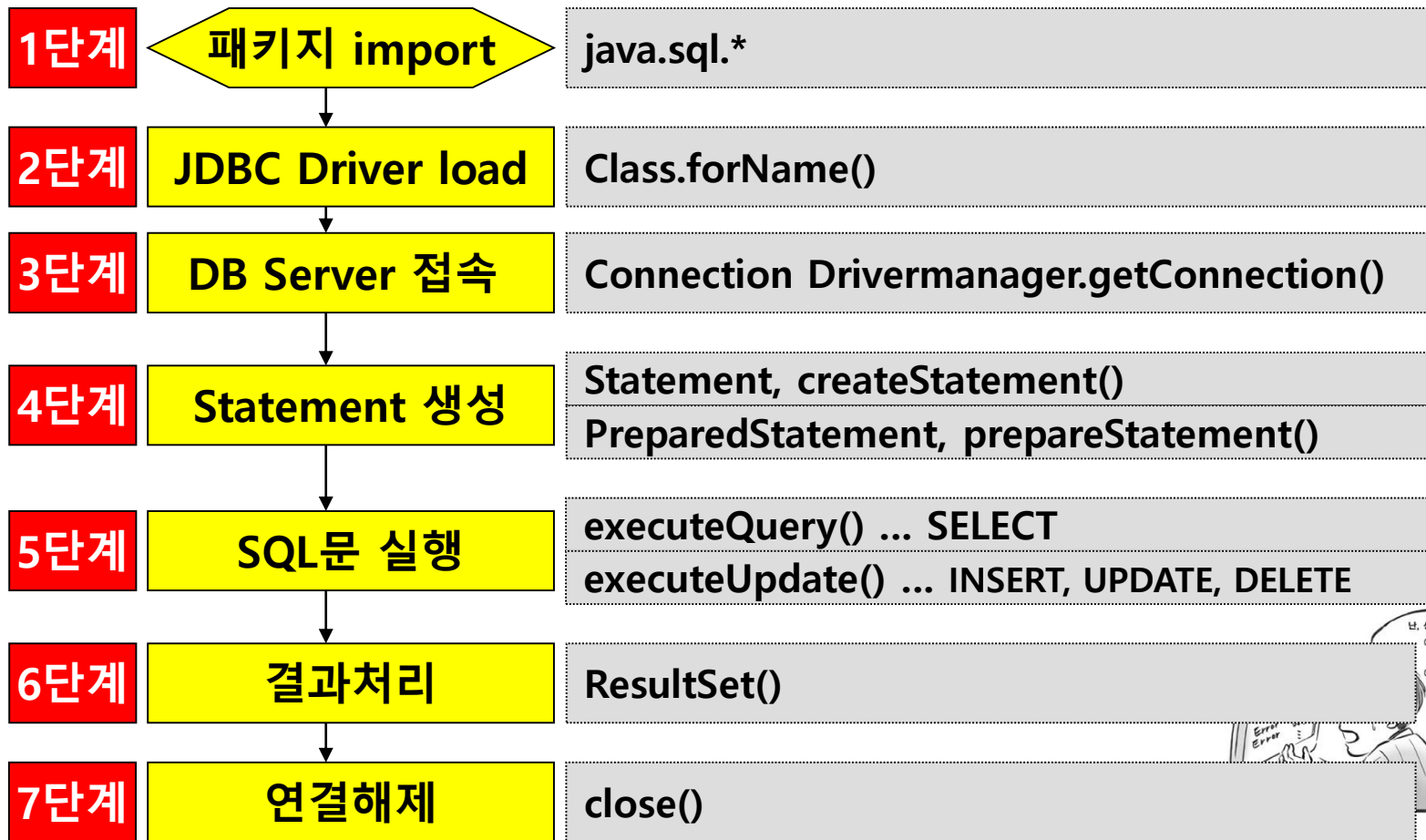
데이터 삭제

• DELETE 문

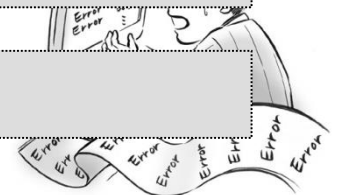


# 복습하기

## ▶ JDBC 프로그래밍 절차



난, 생각하는게 지겨워!  
Compile, Debug, Run, Fix  
또  
Compile, Debug, Run, Fix



## Part2. 기초 프로그래밍

- ▶ Chapter4. JSP 기본문법
- ▶ Chapter5. JSP 내장객체
- ▶ Chapter6. JSP 입력 폼 설계
- ▶ Chapter7. JSP와 DB 연동
- ▶ Chapter8. 자바빈과 액션태그
- ▶ Chapter9. 쿠키와 세션
- ▶ Chapter10. 서블릿
- ▶ Chapter11. DBCP
- ▶ Chapter12. EL
- ▶ Chapter13. JSTL

문법과  
기초 프로그래밍 실습

# 오늘의 수업 주제



자바빈



자바빈의 표준액션태그



부가적인 표준액션태그

# 강의 목표

1. 자바빈의 이해와 생성 방법 익히기
2. 표준액션태그를 이용하여 자바빈 활용하기
  - ▶ 3종류
3. 부가적인 표준액션태그의 종류와 활용
  - ▶ 4종류



## 8. 자바빈과 표준액션태그

1. 자바빈(JavaBeans)
2. 자바빈의 표준액션태그
3. 추가적인 표준액션태그



# JSP 액션 태그

## ▶ JSP 표준액션태그

- ▶ 클라이언트 또는 웹서버에게 어떠한 동작을 지시하는 태그
- ▶ 자바빈즈 지원 태그 : 3종류
- ▶ 부가적인 기능 지원 태그 : 4종류

종 류	설 명
<jsp:useBean>	자바빈을 JSP 페이지와 연결시킬 때
<jsp:setProperty>	자바빈의 속성 값을 할당
<jsp:getProperty>	자바 빈의 속성 값을 얻음
<jsp:include>	정적 또는 동적인 페이지를 현재 페이지에 포함시킴
<jsp:forward>	다른 JSP 페이지 또는 사이트로 이 동
<jsp:plugin>	웹 브라우저에서 자바 플러그인(plugin)을 사용케 함
<jsp:param>	<jsp:include>, <jsp:forward>, <jsp:plugin>의 서브 원 소, 키와 값을 전달할 때 사용

# 1. 자바빈

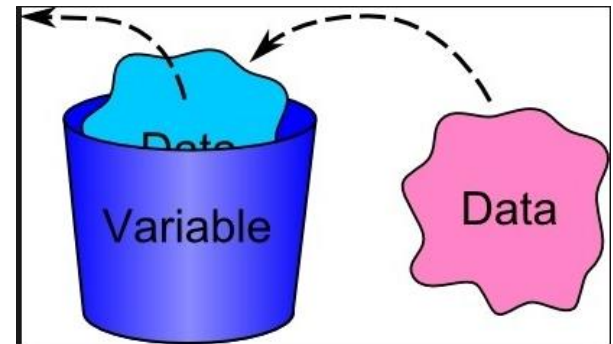


## ▶ 자바빈(JavaBeans)?

- ▶ 재사용이 가능한 객체지향 프로그래밍 인터페이스
- ▶ 데이터를 다루기 위해서 자바로 작성되는 소프트웨어 컴포넌트
- ▶ JSP에서 사용하는 자바빈의 컴포넌트 클래스
- ▶ 구조 단순, 작성 용이

## ▶ JSP 페이지에서 자바빈

- ▶ 입력 폼의 데이터와 데이터베이스의 데이터 처리 부분에 활용
- ▶ 재사용이 가능한 객체
- ▶ MVC 설계 패턴에서 모델로 사용





# 1. 자바빈

## ▶ 1) 자바빈의 규약

- ▶ 클래스는 반드시 파라메타가 없는 기본 생성자를 가진다.
- ▶ 클래스는 직렬화되어야 한다.
- ▶ 프로퍼티는 get, set 메서드를 통해 읽고 저장할 수 있어야 한다.
- ▶ 프로퍼티마다 Getter, Setter가 있어야 한다.
- ▶ Getter는 파라메타가 없어야 한다.
- ▶ Setter는 하나 이상의 파라메타를 가진다.
- ▶ 프로퍼티 접근제어자는 private, Getter/Setter와 클래스는 public으로 선언해야 한다.



# 1.1 자바빈의 구성



## ▶ 자바빈

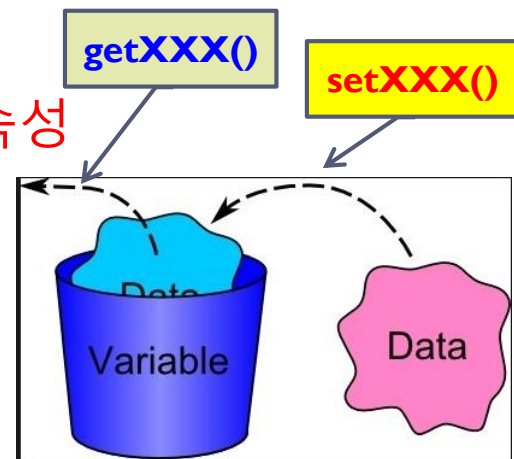
- ▶ 클래스(class)로 선언
- ▶ 클래스는 프로퍼티(property)와 메서드(method)로 구성

## ▶ 프로퍼티

- ▶ 클래스의 상태를 유지하고, 동작을 지정하는 속성
- ▶ 변수형태
- ▶ 예: 입력폼의 필드명, 테이블의 칼럼명 등

## ▶ 메서드

- ▶ 프로퍼티를 외부에 사용하기 위한 통로 역할
- ▶ 컴포넌트를 조작할 수 있도록 하는 방식을 제공
- ▶ 함수 형태

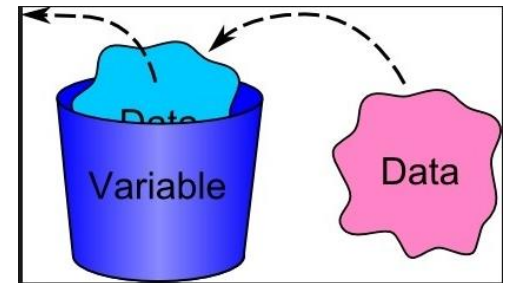


# 1.1 자바빈의 구성



## ▶ 일반형식

```
접근제어자 class 클래스명 {  
    접근제어자 데이터타입 프로퍼티명 ;  
    ...  
    접근제어자 데이터타입 메서드명() {  
        ...  
    }  
}
```



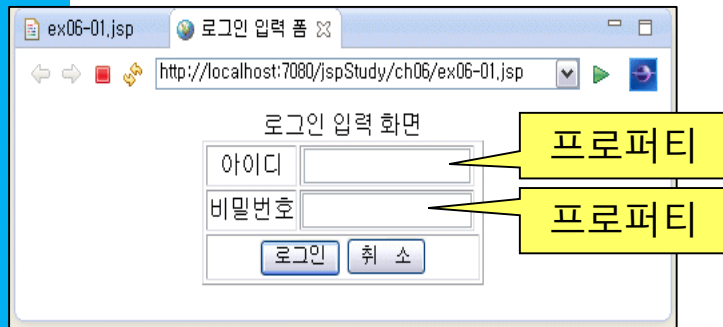
## ▶ 접근제어자

- ▶ **public**이나 **private** 등 클래스에서 각 프로퍼티나 메서드 앞에 기술하는 예약어
- ▶ 클래스 내부에 선언된 프로퍼티와 메서드의 접근 권한을 제한
  - **public** : 외부의 접근을 허용, 클래스나 메서드 선언시 사용
  - **private** : 클래스내에서만 접근 허용, 프로퍼티 선언시 사용

## ▶ 데이터타입

- ▶ 자바문법에 정의하는 데이터 타입 기술

# 1.2 입력 폼의 자바빈 작성



## 로그인 입력화면의 프로퍼티

### 데이터를 입력하는 필드

- 아이디(id), 비밀번호(pw)

### 프로퍼티 선언

- `private String id = "초기값";`
- `private String pw = "초기값";`

## 메서드 선언

### 하나의 프로퍼티에 대하여

- `setXXX()`, `getXXX()` 메서드가 쌍으로 선언

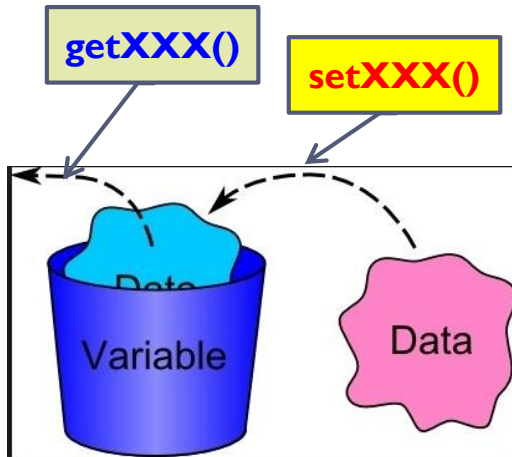
- `setXXX()`

- 프로퍼티에 값을 저장, 전달인자가 프로퍼티 전달인자와 동일한 데이터타입, 반환 값이 없음

- `getXXX()`

- 프로퍼티 값을 반환, 전달인자가 없음, 반환 값이 프로퍼티와 동일한 데이터타입

- `XXX`의 메서드명은 프로퍼티명으로 첫 문자를 대문자로 지정



# 1.2 입력 폼의 자바빈 작성

- 아이디(id) 프로퍼티를 위한 메서드 선언

```
public void setId(String id) {
```

```
    this.id = id;
```

반환 값이 없는 타입으로 선언

```
public String getId() {
```

```
    return id;
```

```
}
```

- 클래스 선언

```
public class LoginBean {
```

```
    ...
```

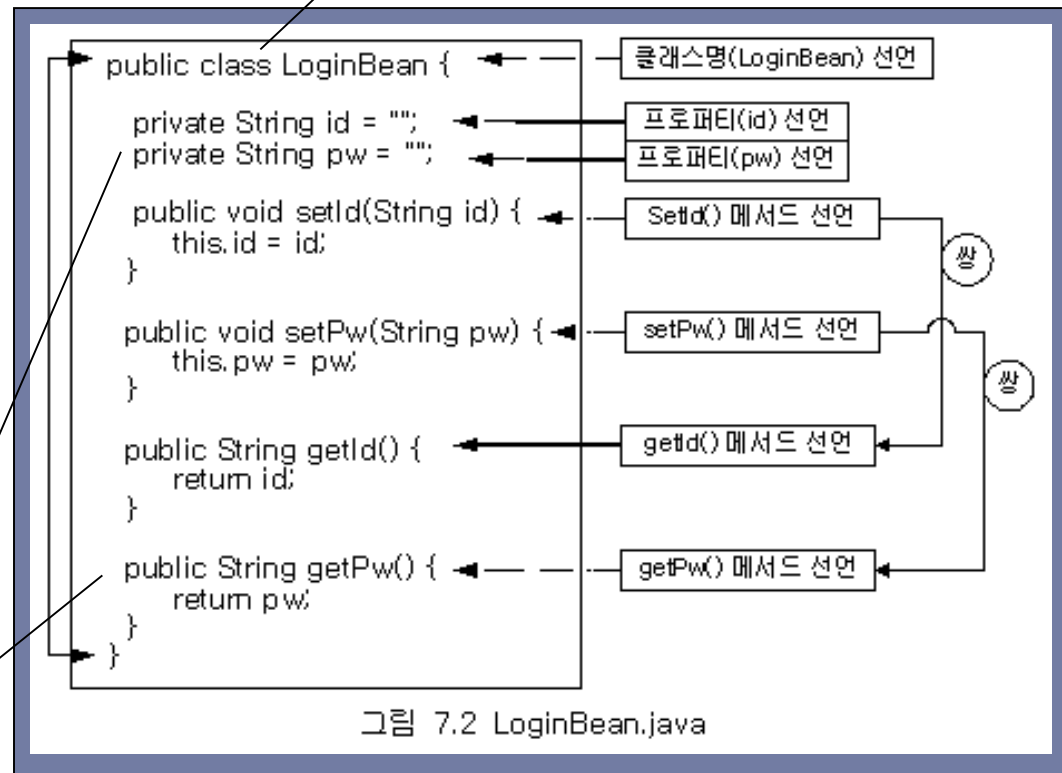
```
}
```

id의 반환 값

프로퍼티는 메서드에서만 접근  
가능하도록 private 사용

클래스 외부에서 인터페이스가 가능하도록  
하기 위해서 메서드는 public 사용

class 키워드는 클래스 정의



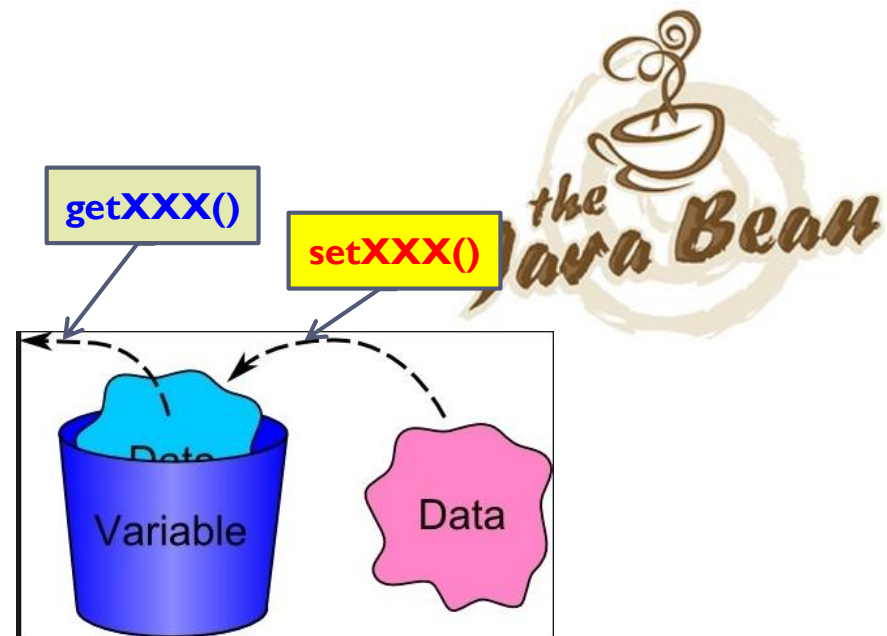
## 예제 8.1

- ▶ 그림 8.1의 로그인 입력화면을 이용하여 LoginBean의 자바빈을 코딩하시오. 단, id와 pw 프로퍼티의 초기값은 공백이다.

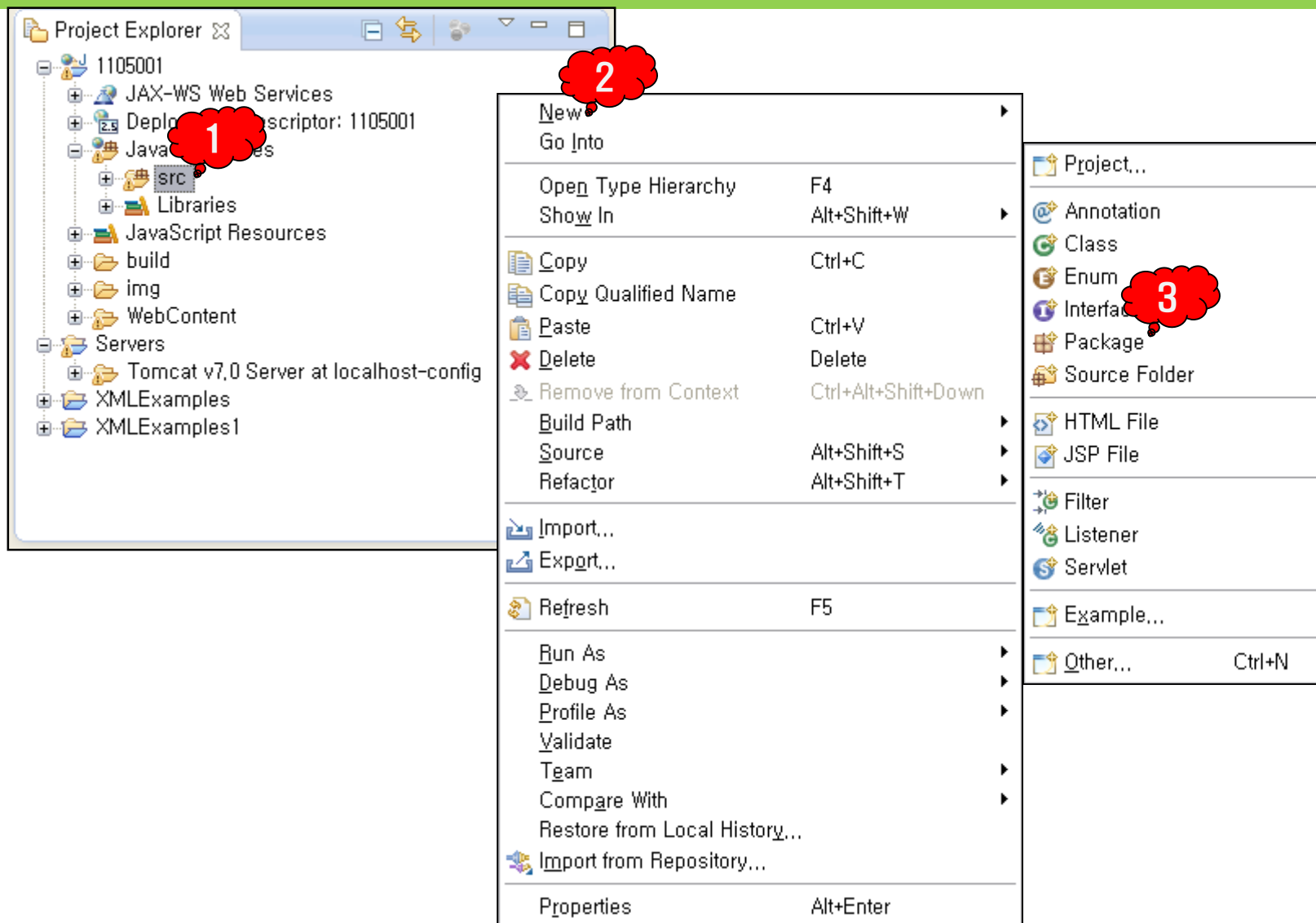
```
1 package ch08;  
2  
3 public class LoginBean {  
4     private String id = "";  
5     private String pw = "";  
6  
7     public String getId() {  
8         return id;  
9     }  
10    public void setId(String id) {  
11        this.id = id;  
12    }  
13    public String getPw() {  
14        return pw;  
15    }  
16    public void setPw(String pw) {  
17        this.pw = pw;  
18    }  
19  
20 }
```

package 패키지명;

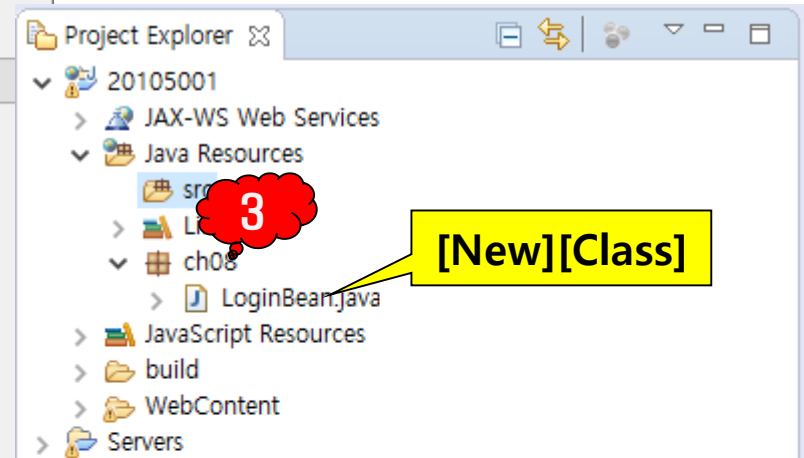
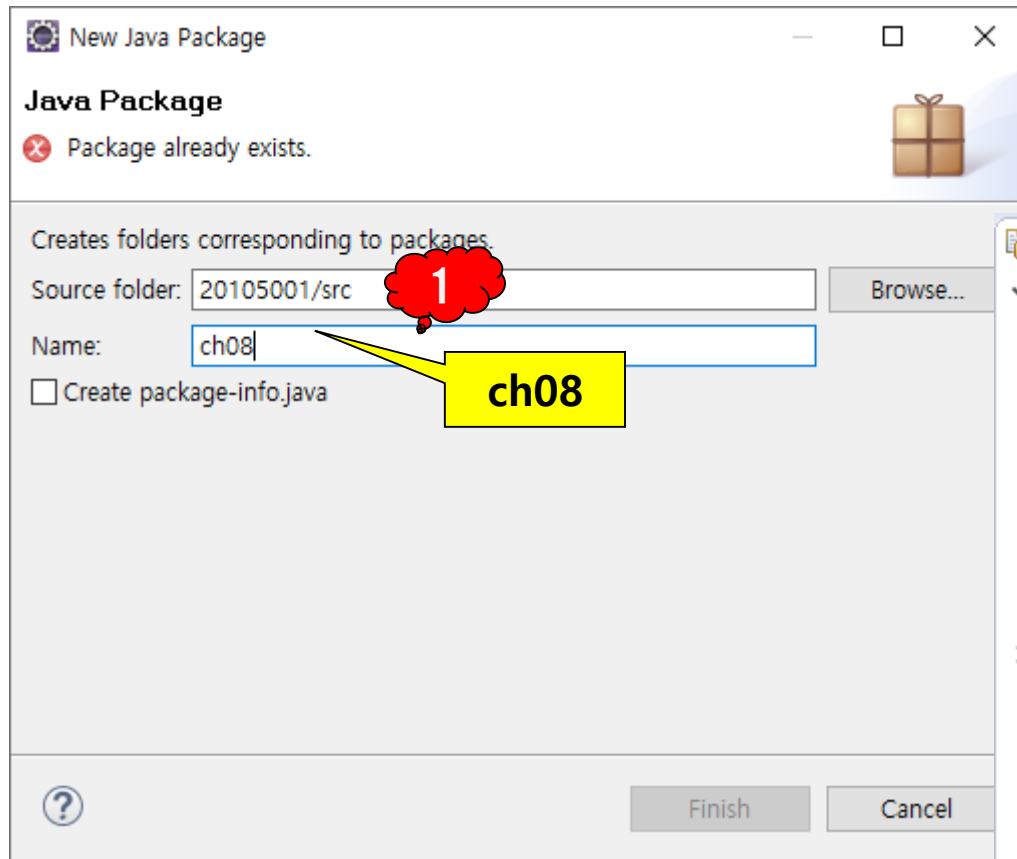
- 자바에서 기본적으로 사용되는 구조의 개념
- 자바 소스 프로그램의 맨 윗줄에 선언



# 1.3 이클립스에서 자바빈 입력



# 1.3 이클립스에서 자바빈 입력





# 1.3 이클립스에서 자바빈 입력

New Java Class

Java Class

✖ Type already exists.

Source folder: 20105001/src **ch08** Browse...

Package: ch08 Browse...

☐ Enclosing type: Browse...

Name: LoginBean **1**

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

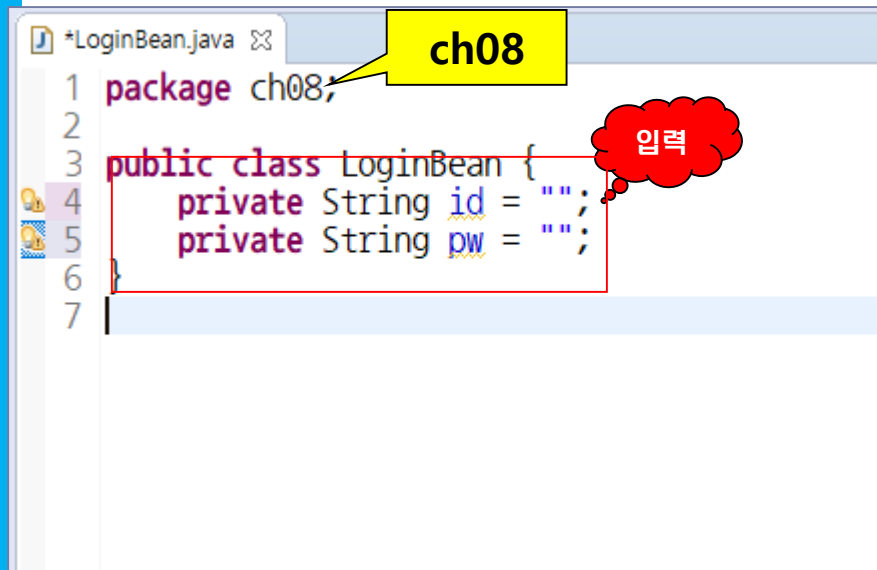
Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

Finish Cancel

LoginBean.java **ch08**

```
1 package ch08;  
2  
3 public class LoginBean {  
4  
5 }  
6
```

# 1.3 이클립스에서 자바빈 입력



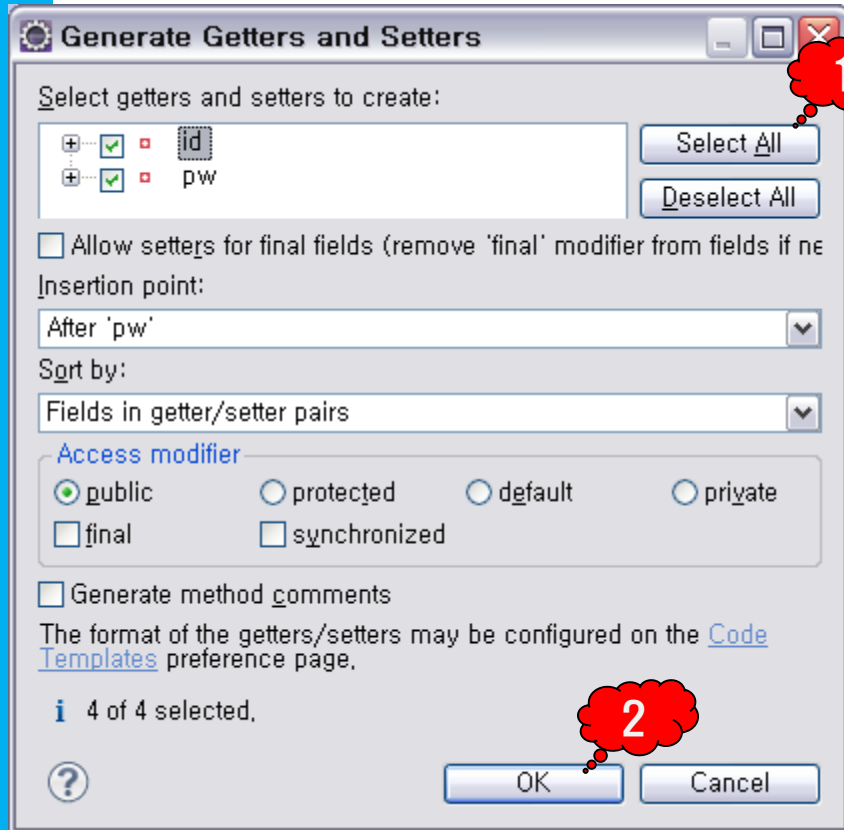
```
1 package ch08;
2
3 public class LoginBean {
4     private String id = "";
5     private String pw = "";
6 }
7
```

## [Source] 메뉴

Toggle Comment	Ctrl+7
Add Block Comment	Ctrl+Shift+/*
Remove Block Comment	Ctrl+Shift+*/
Generate Element Comment	Alt+Shift+J
Shift Right	
Shift Left	
Correct Indentation	Ctrl+I
Format	Ctrl+Shift+F
Format Element	
Add Import	Ctrl+Shift+M
Organize Imports	Ctrl+Shift+O
Sort Members...	
Clean Up...	
Override/Implement Methods...	
Generate Getters and Setters...	
Generate Delegate Methods...	
Generate toString()...	
Generate hashCode() and equals()...	
Generate Constructor using Fields...	
Generate Constructors from Superclass...	
Surround With	Alt+Shift+Z ▶
Externalize Strings...	
Find Broken Externalized Strings	

메서드는 메뉴로 생성

# 1.3 이클립스에서 자바빈 입력



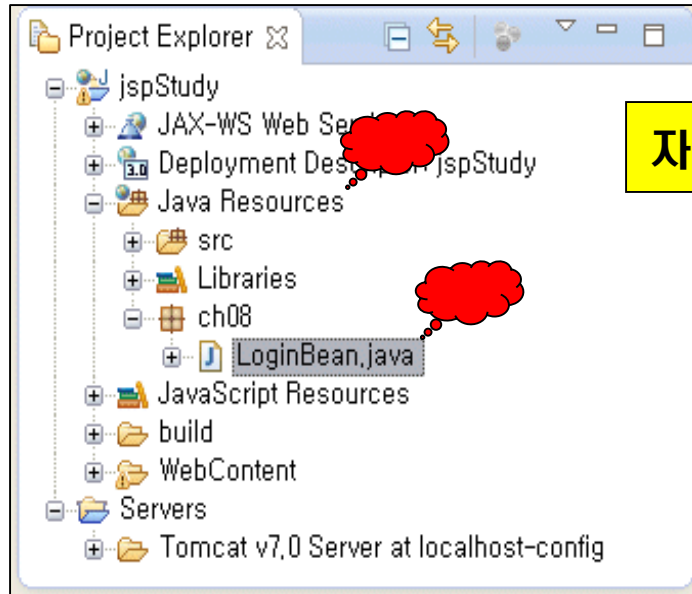
ch08

```
1 package ch08;
2
3 public class LoginBean {
4     private String id = "";
5     private String pw = "";
6
7     public String getId() {
8         return id;
9     }
10    public void setId(String id) {
11        this.id = id;
12    }
13    public String getPw() {
14        return pw;
15    }
16    public void setPw(String pw) {
17        this.pw = pw;
18    }
19 }
20 }
```

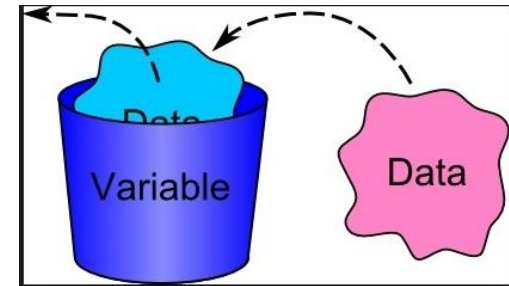
자동생성



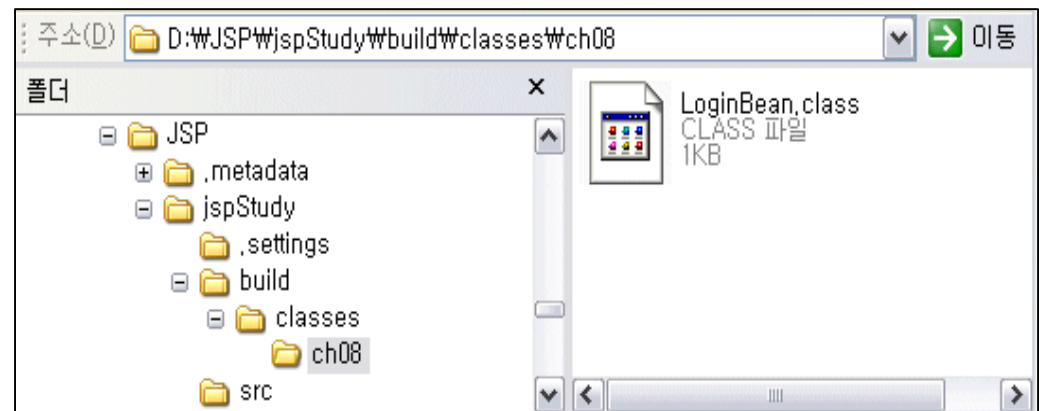
# 1.3 이클립스에서 자바빈 입력



자바빈 생성위치

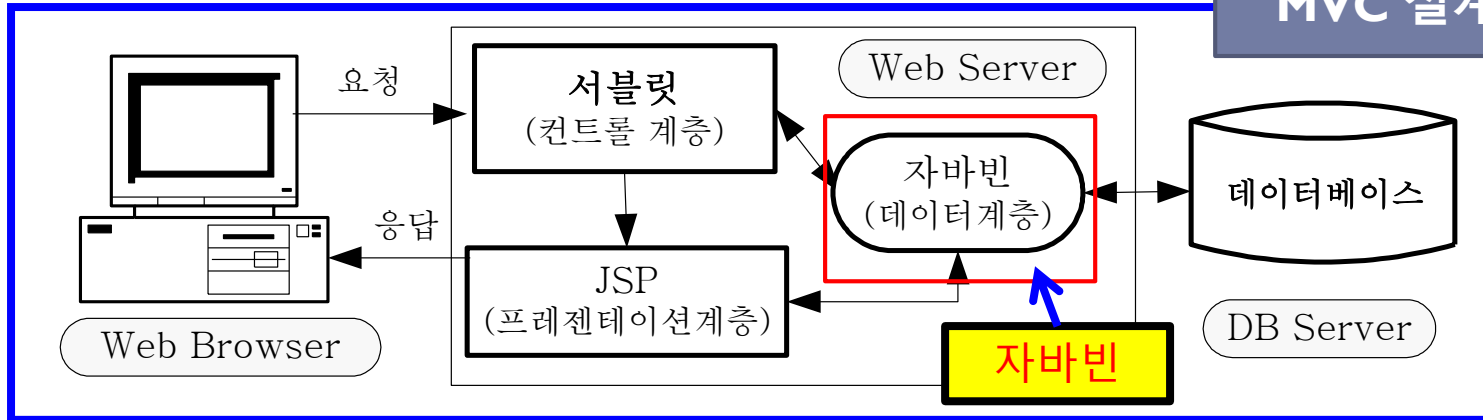


자동 컴파일. \*.class 생성



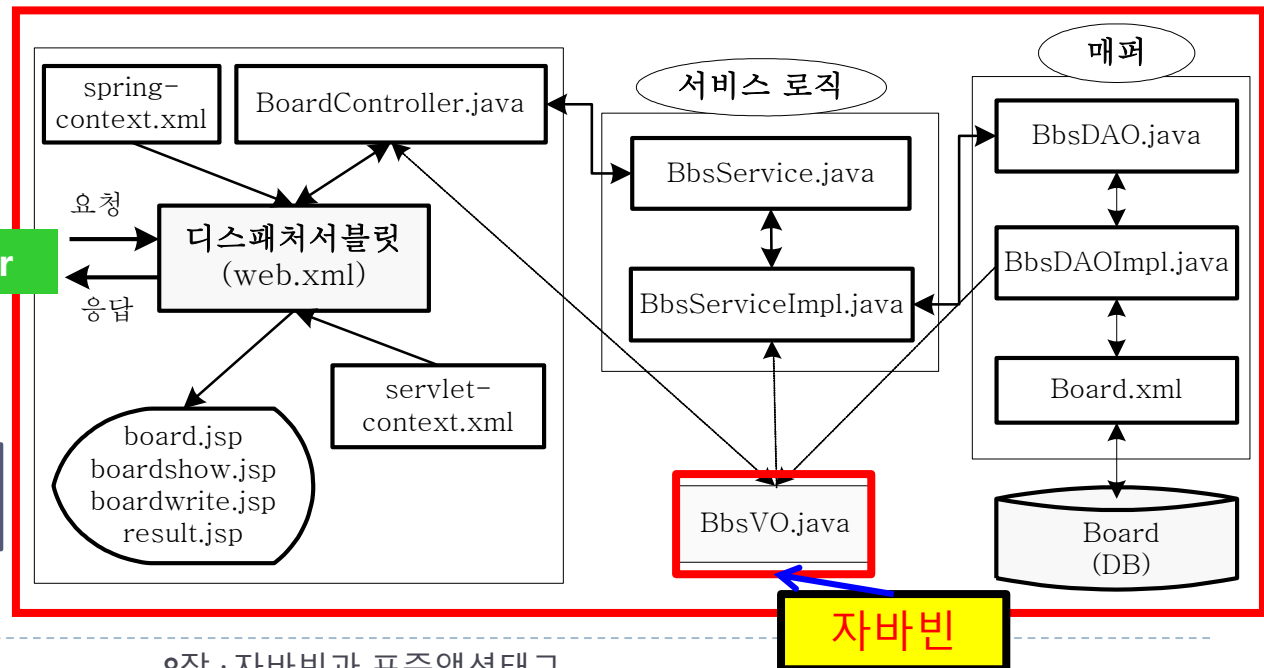
# 웹프로젝트 개발의 자바빈 활용

## MVC 설계



## Web Browser

## 스프링웹MVC +MyBatis

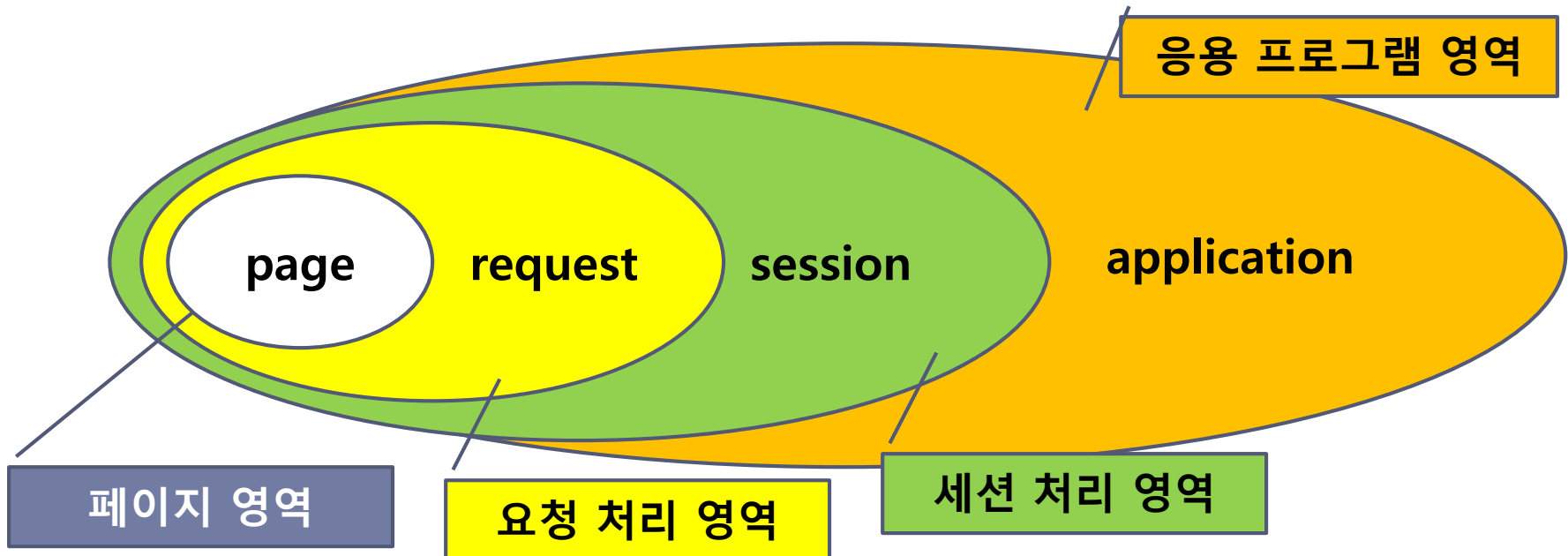


## 자바빈

## 2. 자바빈 표준 액션 태그

### ▶ 자바빈의 스코프(scope) 속성

영역	설명
page	요청에 의해 생성된 페이지에서만 객체가 사용
request	클라이언트의 요청이 수행하는 페이지에서 객체가 사용
session	객체가 생성된 세션에서 요청을 처리하는 페이지에 사용
application	객체가 생성된 응용 프로그램에 포함된 페이지에서 사용



## 2.1 자바빈 표준 액션 태그

### 1. <jsp:useBean> 태그

- ▶ JSP 페이지에서 주어진 영역에 특정 자바빈을 사용하기 위해 컨테이너에게 알려주는 태그
- ▶ 자바빈 이름 지정

### 2. <jsp:setProperty> 태그

- ▶ JSP 페이지에서 <jsp:useBean> 태그로 지정한 자바빈에 주어진 영역에서 자바빈의 프로퍼티 값 설정

### 3. <jsp:getProperty> 태그

- ▶ JSP 페이지에서 <jsp:useBean> 태그로 지정한 자바빈으로부터 주어진 영역에서 자바빈의 프로퍼티 값을 반환



# <jsp:useBean> 액션 태그

## 표기법

```
<jsp:useBean id="name"  
             scope="page | request | session | application"  
             class="className"  
             type="typeName" beanname="beanName" />
```

속성	설 명
id	JSP 페이지에서 빈즈의 참조이름을 지정
scope	빈즈의 유효 범위를 지정, 디폴트는 page
type	빈즈의 타입을 기술
class	자바 빈즈의 실제 클래스 명을 기술
beanName	class 속성을 지정하지 않을 경우, 빈즈의 이름으로 지정

예:

```
<%@ page import="ch08.LoginBean" %>
```

```
<jsp:useBean id="test" class="ch08.LoginBean" scope="page" />
```



# <jsp:setProperty> 액션 태그

## 표기법

```
<jsp:setProperty name="beanName"  
    property="propertyName" |  
    property="propertyName" param="parameterName" |  
    property="propertyName" value="propertyValue" />
```

속성	설 명
name	<jsp:useBean>의 id속성에 정의된 빈의 이름
property	빈의 속성 이름
param	request parameter의 이름
value	property에 할당할 값

예:

```
<%@ page import="ch08.LoginBean" %>
```

```
<jsp:useBean id="test" class="ch08.LoginBean" scope="page" />
```

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

```
<jsp:setProperty name="test" property="pw" />
```

# <jsp:getProperty> 액션 태그

## 표기법

```
<jsp:getProperty name="beanName" property="propertyName" />
```

속성값	설 명
name	<jsp:useBean>의 id속성에 정의된 빈의 이름
property	값을 가져 올 빈의 속성 이름

예:

```
<%@ page import="ch08.LoginBean" %>
```

```
<jsp:useBean id="test" class="ch08.LoginBean" scope="page" />
```

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

```
<jsp:setProperty name="test" property="pw" />
```

...

```
아 이 디 : <jsp:getProperty name="test" property="id" /> <p>
```

```
비밀번호 : <jsp:getProperty name="test" property="pw" /> <p>
```

...

## 예제 8.2

- ▶ LoginBean을 이용하여 로그인 입력화면에서 입력하고, 이 값들을 LoginBean 자바빈의 값들을 출력하시오.

ex08-02.jsp    빈즈를 사용한 예제

http://localhost:7080/jspStudy/ch08/ex08-02.jsp

로그인 입력 화면

아 이 디 :	<input type="text" value="kdhong"/>
비밀번호 :	<input type="password" value="....."/>
<input type="button" value="로그인"/> <input type="button" value="취 소"/>	



ex08-02.jsp    Login 빈즈 예제 [2]

http://localhost:7080/jspStudy/ch08/ex08-02-1.jsp

로그인 정보

아 이 디 : kdhong

비밀번호 : 12345



```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4   "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6 <head>
7   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8   <title>로그인 입력 화면</title>
9
10  <script language="JavaScript">
11      function In_Check() {
12          if (document.login.id.value == "") {
13              alert("아이디(ID)를 입력하세요!!!");
14              return;
15          }
16          if (document.login.pw.value == "") {
17              alert("비밀번호를 입력하세요!!!");
18              return;
19          }
20          document.login.submit();
21      }
22  </script>
23
24 </head>
25 <body>
26   <center><h4>로그인 입력 화면</h4>
27   <form method="post" action="ex08-02-1.jsp" name="login">
28     <table border="1" cellspacing="1">
29       <tr>
30         <td>아이디: </td>
31         <td><input type="text" name="id" size=15</td>
32       </tr>
33       <tr>
34         <td>비밀번호: </td>
35         <td><input type="password" name="pw" size=17</td>
36       </tr>
37       <tr>
38         <td colspan="2" align="center">
39           <input type="button" value="로그인" onClick="In_Check()">
40           <input type="reset" value="취소">
41         </td>
42       </tr>
43     </table></form>
44   </center>
45 </body>
46 </html>
47

```

ex06-07.jsp 복사후 수정

```
*ex08-02-1.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"
3   import="ch08.LoginBean" %>
4   <jsp:useBean id="test" class="ch08.LoginBean" scope="page" />
5   <jsp:setProperty name="test" property="id" />
6   <jsp:setProperty name="test" property="pw" />
7
8 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
9   "http://www.w3.org/TR/html4/loose.dtd">
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <title>Login 자바빈 예제</title>
14 </head>
15 <body>
16   <h4>로그인 정보</h4>
17   아이디 : <jsp:getProperty name="test" property="id" /><p>
18   비밀번호 : <jsp:getProperty name="test" property="pw" /><p>
19 </body>
20 </html>
```

## 2.4 자바빈 사용시 주의사항

HTML 입력 폼

입력화면

```
<BODY>
<center><h3> 로그인 입력 화면 </h3>
<form method="post" action="ex7-02.jsp" name="login">
  <table border="1" cellspacing="1">
    <tr>
      <td>아 이 디 : </td>
      <td><input type="text" name="id" size=10></td>
    </tr>
    <tr>
      <td>비밀번호 : </td>
      <td><input type="password" name="pw" size=10></td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <input type="button" name="confirm" value="로그인" onClick="In_Check()">
        <input type="reset" name="reset" value="취 소">
      </td>
    </tr>
  </table></form></center>
```

자바빈

```
package ch08;

public class LoginBean {

  private String id = "guest";
  private String pw = "12345";

  public void setId(String id) {
    this.id = id;
  }
  public void setPw(String pw) {
    this.pw = pw;
  }
  public String getId() {
    return id;
  }
  public String getPw() {
    return pw;
  }
}
```

JSP 프로그램

```
<%@ page import="ch8.LoginBean" %>
<jsp:useBean id="test" class="ch8.LoginBean" scope="page" />
<jsp:setProperty name="test" property="id" />
<jsp:setProperty name="test" property="pw" />

<HTML>
  <HEAD><TITLE> LoginBean빈즈를 사용하는 예제(2) </TITLE>
  </HEAD>
  <BODY>
    <h3> LoginBean 빈즈 예제(2) </h3>
    아 이 디 : <%= test.getId() %><p>
    비밀번호 : <%= test.getPw() %><p>
    로 로그인 되었습니다.
  </BODY></HTML>
```

# 엔터프라이즈 자바빈 (EJB)

- ▶ 엔터프라이즈 자바빈(Enterprise JavaBeans; EJB)
  - ▶ 기업환경의 시스템을 구현하기 위한 서버측 컴포넌트 모델
  - ▶ 애플리케이션의 업무 로직을 가지고 있는 서버 애플리케이션
  - ▶ 웹 애플리케이션에서 JSP는 화면 로직을 처리하고, EJB는 업무 로직을 처리하는 역할
- ▶ EJB의 3가지 종류
  - ▶ 세션 빈(Session Bean)
  - ▶ 데이터베이스와 연동하여 데이터를 처리하는 엔티티 빈(Entity Bean)
  - ▶ JMS로 빈을 날려주는 메시지 구동 빈 (Message-driven Bean)

### 3. 추가적인 액션 태그

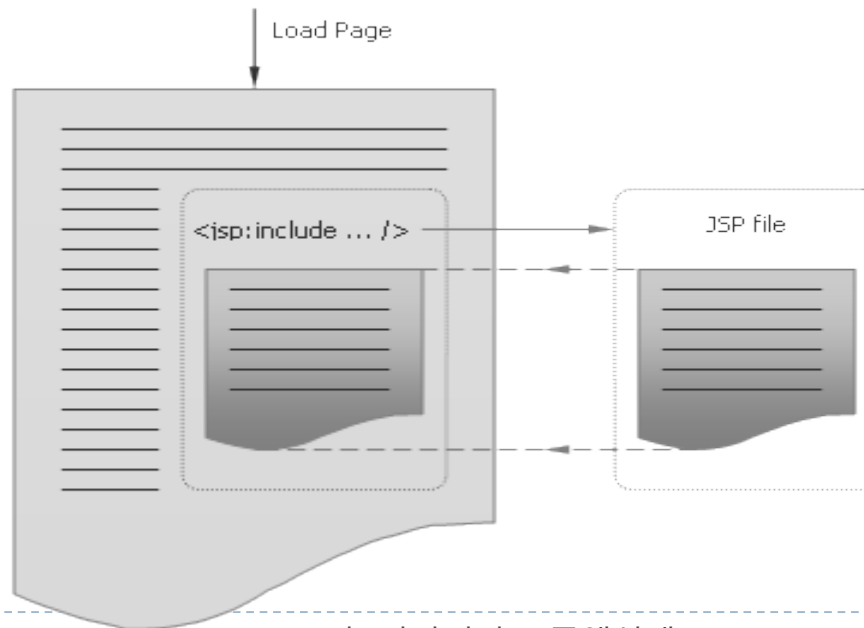
- ▶ `<jsp:include>` 태그
  - ▶ JSP 페이지에서 다른 JSP 페이지나 HTML 페이지를 포함하는 태그
- ▶ `<jsp:forward>` 태그
  - ▶ 현재 실행중인 JSP 페이지에서 url로 지정한 특정 JSP 페이지로 이동하는 액션 태그
- ▶ `<jsp:param>` 태그
  - ▶ 요청한 페이지로 정보를 전달할 때 사용하는 액션 태그



# 3.1 <jsp:include> 액션 태그

## ▶ <jsp:include> 태그

- ▶ JSP 페이지에서 다른 JSP 페이지나 HTML 페이지를 포함하는 태그
- ▶ 주로 동적인 웹 페이지를 포함
- ▶ 정적인 웹페이지 문서도 가능
- ▶ 헤더 정보나 쿠키 값 설정이 불가능하고, 페이지 변경이 발생할 때마다 컴파일하기 때문에 성능을 저하시키는 요인이 발생



## 3.1 <jsp:include> 액션 태그

**표기법** <jsp:include page="포함시킬 파일명" flush="true 혹은 false" />

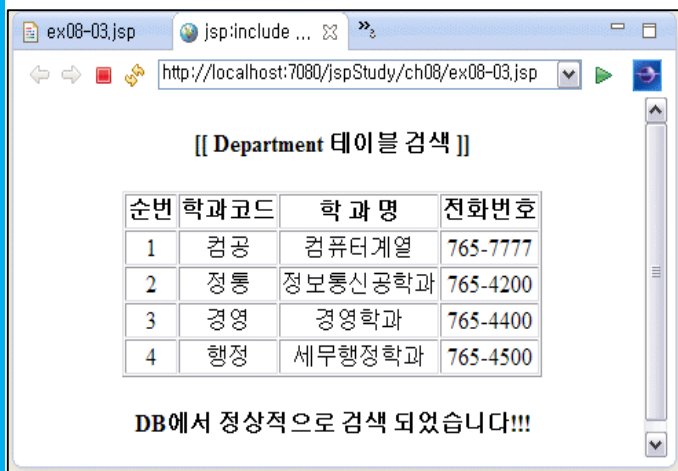
속성	설 명
page	포함시킬 페이지의 상대 url
flush	true 기술시 버퍼가 flush

구 분	<jsp:include> 액션 태그	include 지시어
사용형식	<jsp:include page="..." />	<%@ include file="..." %>
포함 시점	실행시	컴파일시
지원 자원	정적, 동적인 페이지	정적인 페이지

**<jsp:include> 태그와 include 지시어와의 차이점**

## 예제 8.3

- ▶ 예제 8.3 출력 결과와 같이 Department 테이블을 조회하는 프로그램(ex07-01.jsp)을 이용하여 작성하시오.



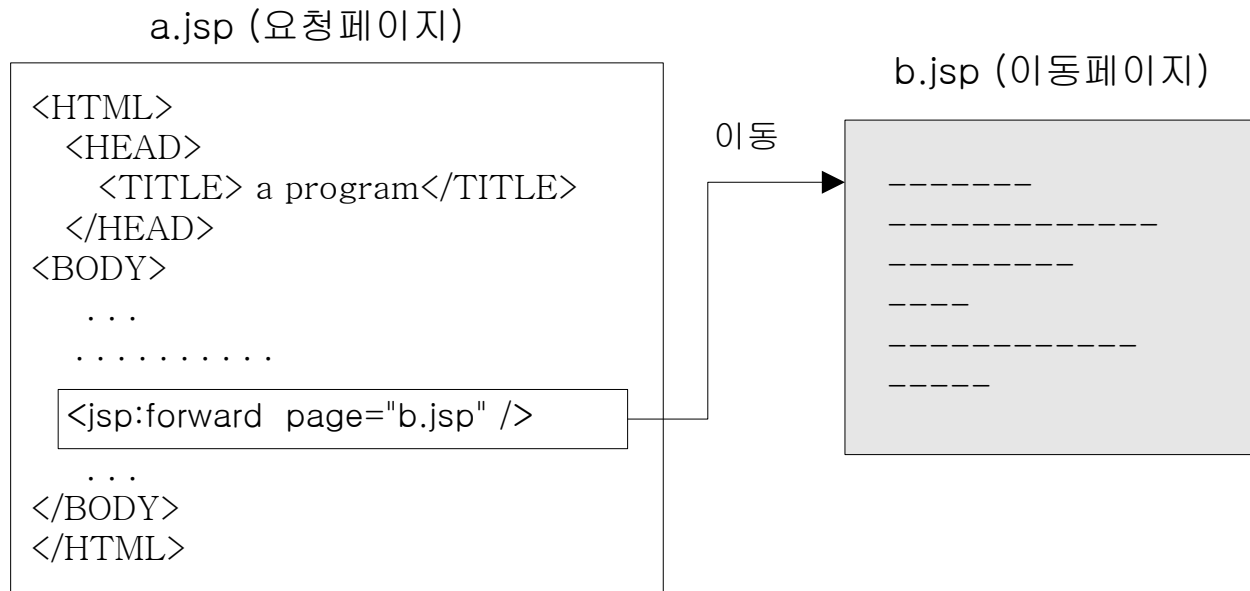
```
ex08-03.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>jsp:include 태그 예제</title>
8 </head>
9 <body>
10 <jsp:include page="../ch07/ex07-01.jsp" flush="true" />
11 </body>
12 </html>
```

## 3.2 <jsp:forward> 액션 태그

### ■ <jsp:forward> 태그

- 현재 실행중인 JSP 페이지에서 url로 지정한 특정 JSP 페이지로 이동하는 액션 태그

**표기법**    `<jsp:forward page="fileurl" />`



## 3.3 <jsp:param> 액션 태그

### ▶ <jsp:param>태그

- ▶ 요청한 페이지로 정보를 전달할 때 사용하는 액션 태그
- ▶ 단독 사용 불가
- ▶ <jsp:include>, <jsp:forward>, <jsp:plugin> 서브원소로 사용
- ▶ 현재 페이지에서 요청한 페이지로 파라미터 값을 전달

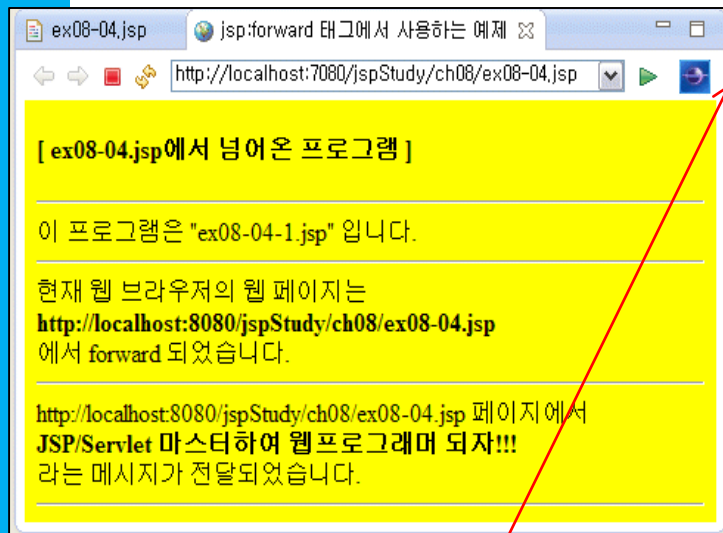
표기법	<code>&lt;jsp:param name="name" value="값" /&gt;</code>
-----	--

속성	설 명
name	파라미터의 이름 기술
value	파라미터 값을 기술

```
예: <jsp:forward page="urlfile"
      <jsp:param name="name" value="값" />
    </jsp:forward>
```

## 예제 8.4

- ▶ <jsp:forward> 액션 태그를 이용하여 "JSP/Servlet 마스터로 웹프로그래머가 되자"라는 정보를 ex08-04-1.jsp로 전달하고, 예제 8.4의 출력 결과와 같이 출력하는 JSP 프로그램을 작성하라.



```
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8 <title>jsp:forward 태그에서 사용하는 예제</title>
9 </head>
10 <body bgcolor="yellow">
11     <h4> [ ex08-04.jsp에서 넘어온 프로그램 ]</h4>
12     <hr>이 프로그램은 "ex08-04-1.jsp" 입니다. <br><hr>
13     현재 웹 브라우저의 웹 페이지는 <br>
14     <b> <%= request.getParameter("url") %> </b> <br>
15     에서 forward 되었습니다. <br><hr>
16     <b> <%= request.getParameter("url") %> 페이지에서 <br>
17     <b> <%= request.getParameter("memo") %> </b><br>
18     라는 메시지가 전달되었습니다.</b>
19 </body>
20 </html>
```

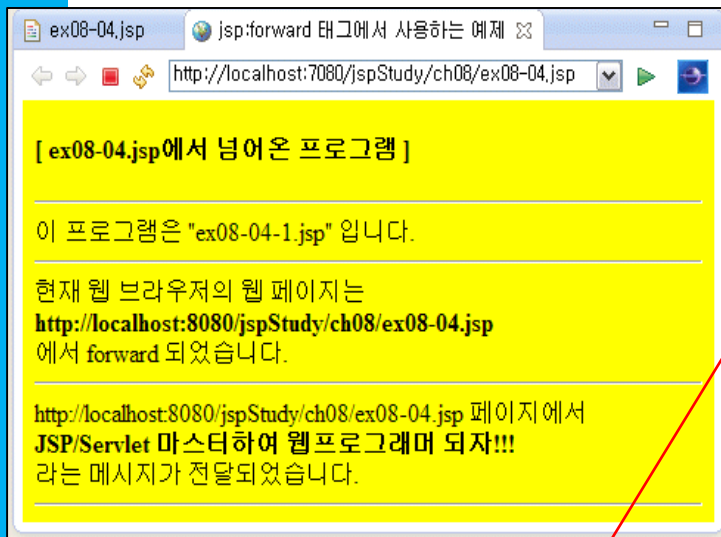
ex08-04-1.jsp

```
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8 <title>jsp:forward와 jsp:param 태그 예제</title>
9 </head>
10 <body bgcolor="red">
11     <H4> [jsp:forward와 jsp:param 태그에 대한 예제] </H4>
12     <hr>이 파일은 ex08-04.jsp입니다. 배경색은 빨간색입니다.
13     <jsp:forward page="ex08-04-1.jsp">
14         <jsp:param name="url"
15             value="http://localhost:8080/jspStudy/ch08/ex08-04.jsp" />
16         <jsp:param name="memo" value="JSP/Servlet 마스터하여 웹프로그래머 되자!!!" />
17     </jsp:forward>
18 </body>
19 </html>
```

ex08-04.jsp

## 예제 8.4

- ▶ <jsp:forward> 액션 태그를 이용하여 "JSP/Servlet 마스터로 웹프로그래머가 되자"라는 정보를 ex08-04-1.jsp로 전달하고, 예제 8.4의 출력 결과와 같이 출력하는 JSP 프로그램을 작성하라.



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <% request.setCharacterEncoding("UTF-8"); %>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
5     "http://www.w3.org/TR/html4/loose.dtd">
6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
9 <title>jsp:forward 태그에서 사용하는 예제</title>
10 </head>
11 <body bgcolor="yellow">
12 <h4>[ ex08-04.jsp에서 넘어온 프로그램 ]</h4>
13 <hr>이 프로그램은 "ex08-04-1.jsp" 입니다. <br><hr>
14 현재 웹 브라우저의 웹 페이지는 <br>
15 <b>%=request.getParameter("url") %</b> <br>
16 에서 forward 되었습니다.<br><hr>
17 <b>%=request.getParameter("url") %</b> 페이지에서<br>
18 <b>%= request.getParameter("memo") %</b><br>
19 라는 메시지가 전달되었습니다.<hr>
20 </body>
21 </html>
```

ex08-04-1.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <% request.setCharacterEncoding("UTF-8"); %>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8 <title>jsp:forward와 jsp:param 태그 예제</title>
9 </head>
10 <body bgcolor="red">
11 <h4>[jsp:forward와 jsp:param 태그에 대한 예제]</h4>
12 <hr>
13 이 파일은 ex08-04.jsp 입니다. 배경색은 빨간색 입니다.
14 <jsp:forward page="ex08-04-1.jsp">
15     <jsp:param name="url" value="http://localhost:8080/jspStudy/ch08/ex08-04-1.jsp" />
16     <jsp:param name="memo" value="JSP/Servlet 마스터하여 웹프로그래머 되자!!!" />
17 </jsp:forward>
18 </body>
19 </html>
```

ex08-04.jsp

# 강 의 내 용 요약 정 리

## ▶ 자바빈

- ▶ 클래스(class)로 선언
- ▶ 클래스는 프로퍼티(property)와 메서드(method)로 구성

## ▶ 프로퍼티

- ▶ 클래스의 상태를 유지하고, 동작을 지정하는 속성
- ▶ 변수형태
- ▶ 예: 입력폼의 필드명, 테이블의 칼럼명 등

## ▶ 메서드

- ▶ 프로퍼티를 외부에 사용하기 위한 통로 역할
- ▶ 컴포넌트를 조작할 수 있도록 하는 방식을 제공
- ▶ 함수 형태



# 강 의 내 용 요약 정 리

## 표준액션태그 : 7종류

종 류	설 명
<jsp:useBean>	자바빈을 JSP 페이지와 연결시킬 때
<jsp:setProperty>	자바빈의 속성 값을 할당
<jsp:getProperty>	자바 빈의 속성 값을 얻음
<jsp:include>	정적 또는 동적인 페이지를 현재 페이지에 포함시킴
<jsp:forward>	다른 JSP 페이지 또는 사이트로 이 동
<jsp:plugin>	웹 브라우저에서 자바 플러그인(plugin)을 사용케 함
<jsp:param>	<jsp:include>, <jsp:forward>, <jsp:plugin>의 서브 원 소, 키와 값을 전달할 때 사용



장 주제

## 9장. 쿠키와 세션

