

# 학습목표

---

1. 자바빈이 무엇인지를 알고 사용법을 익힌다.
2. 자바빈을 이용하여 Model View Controller를 분리한 코드를 작성할 수 있다.

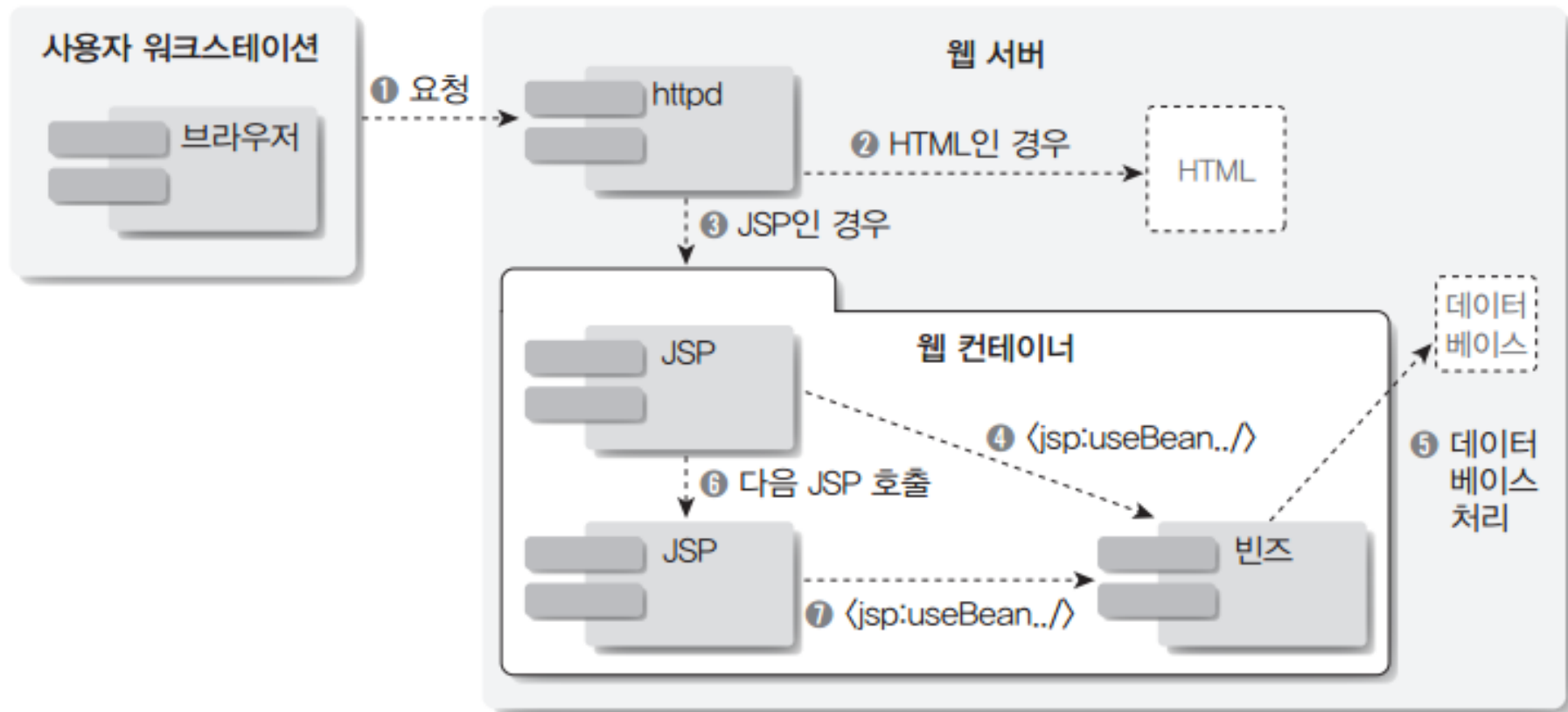
- 자바빈(JavaBean)

---

- 자바빈: 데이터 저장하기 위한 자료구조(클래스)
- JSP와 연동할 수 있음
- 사용 목적
  - 코드의 단순화와 재사용성을 위함
  - 로직(Module) 구현 담당

## • 자바빈의 위치

- JSP 빈즈는 JSP와 연동하기 위해 만들어진 컴포넌트 클래스
- 컨테이너에 위치함
- 데이터베이스 연동 등 프로그램적 요소를 모듈화할 수 있도록 함



- 자바빈 클래스 구조

```
class xxxBean {  
    // 멤버변수 : 데이터베이스 테이블의 칼럼 이름과 매칭된다.  
    private String xxx;  
  
    // get, set 메서드 : 멤버변수와 매칭된다.  
    public String getXxx() {  
        return xxx;  
    }  
    public setXxx(String xxx) {  
        this.xxx = xxx;  
    }  
}
```

- JSP에서 자바빈(JavaBean) 사용하기

## 1. 자바빈(JavaBean) 클래스 생성

```
public class Member {  
    private String id;  
    private String pwd;  
    private String name;  
  
    //getter/setter  
    return id;  
}  
public void setId(String id) {  
    this.id = id;  
}  
public String getPwd() {  
    return pwd;  
}  
public void setPwd(String pwd) {  
    this.pwd = pwd;  
}
```

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

# • JSP에서 자바빈(JavaBean) 사용하기

## 2. 회원 가입 양식을 통해 입력된 값을 Member 객체에 저장하고 가져오기

회원가입

아이디	<input type="text" value="smart"/>
비밀번호	<input type="text" value="1234"/>
이름	<input type="text" value="홍길동"/>
	<input type="button" value="회원가입"/>

```
<%
//전송 데이터의 한글 인코딩 방식 지정
request.setCharacterEncoding("UTF-8");
String id = request.getParameter("id");
String pwd = request.getParameter("pwd");
String name = request.getParameter("name");

//Member 객체 생성 및 값 저장
Member member = new Member();
member.setId(id);
member.setPwd(pwd);
member.setName(name);
//member 객체에 저장된 데이터 출력
%>
<h3>회원 정보</h3>
<%= "id: " + member.getId() %> <br>
<%= "pwd: " + member.getPwd() %> <br>
<%= "name: " + member.getName() %> <br>
```

### 회원 정보

id: smart  
pwd: 1234  
name: 홍길동

# • JSP에서 자바빈(JavaBean) 사용하기

## 3. 회원 가입 양식을 통해 입력된 값을 액션 태그로 저장하고 가져오기

회원가입

아이디	<input type="text" value="smart"/>
비밀번호	<input type="text" value="1234"/>
이름	<input type="text" value="홍길동"/>
<input type="button" value="회원가입"/>	

```
<%  
    //전송 데이터의 한글 인코딩 방식 지정  
    request.setCharacterEncoding("UTF-8");  
    //액션 태그 이용한 값 저장 및 가져오기  
%>  
  
<jsp:useBean id="member" class="com.smart.beans.Member" scope="session"/>  
<jsp:setProperty name="member" property="id"/>  
<jsp:setProperty name="member" property="pwd"/>  
<jsp:setProperty name="member" property="name"/>  
<h3>회원 정보</h3>  
  
ID: <jsp:getProperty name="member" property="id" /> <br>  
PW: <jsp:getProperty name="member" property="pwd" /> <br>  
NAME: <jsp:getProperty name="member" property="name" />
```

### 회원 가입

ID: smart  
PW: 1234  
NAME: 홍길동

- useBean 액션 태그 이용한 자바빈(JavaBean) 사용하기

## 1. 자바빈 객체 생성

```
<jsp:useBean id= "빈 이름" class="자바빈 클래스 이름" [scope="범위"] />
```

```
<%@page import="com.smart.beans.Member" %>
```

```
Member member = new Member();
```

## 2. 자바빈 객체에 프로퍼티 값 저장

```
member.setId("smart");
```

```
<jsp:setProperty name= "빈 이름" property="프로퍼티 이름" [value="프로퍼티에 저장할 값"] />
```

## 3. 자바빈 객체에 저장된 프로퍼티 값 얻기

```
String id = member.getId();
```

```
<jsp:getProperty name= "빈 이름" property="id"/>
```



## • useBean 액션 태그 이용한 자바빈(JavaBean) 사용하기

```
<tr>
<td width="200"> 사용자 ID</td>
<td width="400">
<input type="text" name="id" size="10" maxlength="10">
<input type="submit" value="Send">
</td>
</tr>
```

```
<jsp:useBean id="member" scope="page" class="com.smart.beans.Member">
<jsp:setProperty name="member" property="id"/>
</jsp:useBean>
```

```
public void setId(String id) {
    this.id = id;
}
```

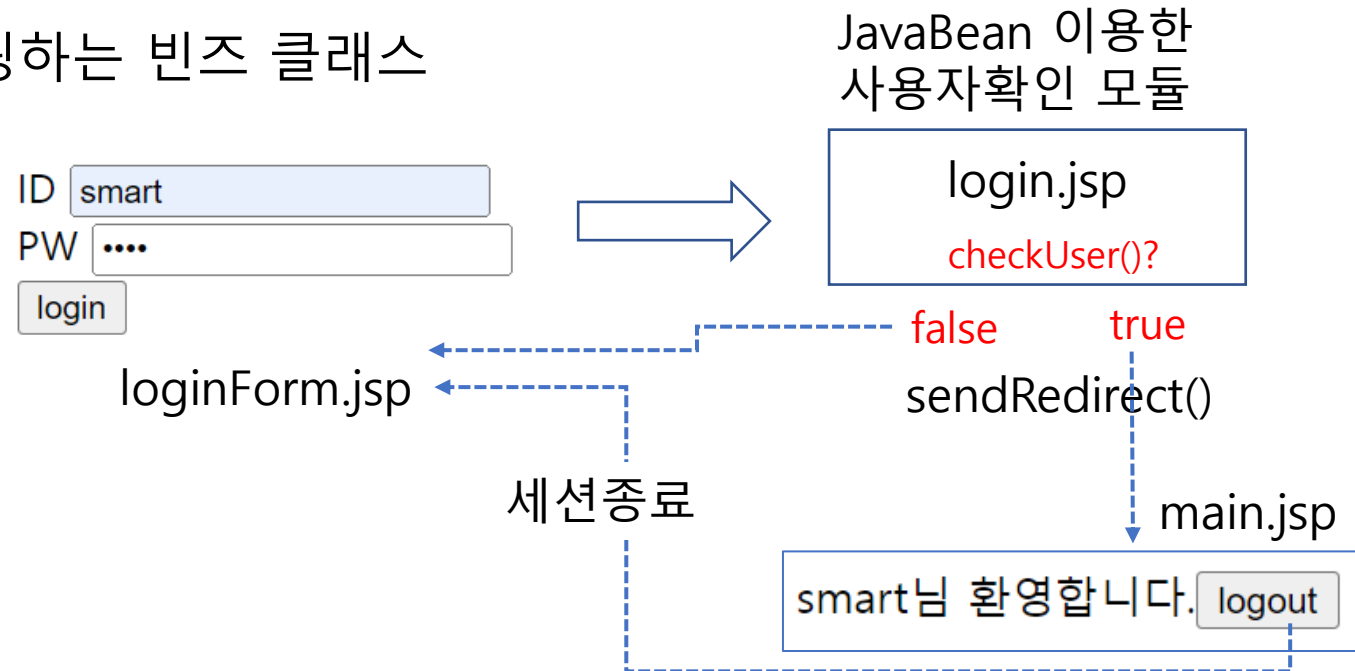
폼 id에 입력된 값이  
액션태그 속성 id에 대입됨

액션태그 속성 id에 대입된  
값이 자바빈 id에 대입됨






# • 자바빈 이용한 사용자 로그인 구현

- LoginForm.jsp
  - 사용자 로그인을 위해 아이디와 비밀번호 입력하는 화면 구현
- login.jsp
  - 입력받은 아이디 정보를 빈즈 클래스를 이용해서 확인하고 처리하는 로직 구현
- Member.java
  - 사용자가 입력한 계정 정보를 매핑하는 빈즈 클래스

```
public class MemberBean {  
    private String id;  
    private String pwd;  
    public MemberBean(){  
        this.id = "smart";  
        this.pwd = "1234";  
    }  
    boolean checkUser(){ }
```



# 실습: 코드 실행과 분석, 수정하기

-  addr\_add.jsp
-  addr\_form.html
-  addr\_list.jsp
-  AddrBean.java
-  AddrManager.java

## 1. 주어진 코드를 실행한 후 분석하기

### 주소록 등록

이름	<input type="text" value="홍길동"/>
전화번호	<input type="text" value="010-2343-2345"/>
이메일	<input type="text" value="kdhong@naver.com"/>
성별	<input type="text" value="남"/>
<input type="button" value="확인"/> <input type="button" value="취소"/>	

addr\_form.html

### 등록내용

이름 : 홍길동  
전화번호 : 010-2343-2345  
이메일 : kdhong@naver.com  
성별 : 남

[목록 보기](#)

addr\_add.jsp

### 주소록

[주소추가](#)

이름	전화번호	이메일	성별
홍길동	010-2343-2345	kdhong@naver.com	남

addr\_list.jsp

# 실습: 코드 실행과 분석, 수정하기

2. 첫페이지(main.jsp) 요청시 "당신은 00번째 방문자입니다."를 출력  
[로그인] [회원가입] 링크 출력
3. AddBean.java 코드에 **grade** 요소 추가(관리자 1, 회원 0 값을 갖는다.)  
=> 회원가입 폼 수정
4. 로그인 id, pwd => 관리자 모드인지 확인  
관리자 모드이면 회원 목록 보기와 주소 추가 가능  
일반회원 모드이면 회원 목록 보기 가능
5. 스크립트릿을 액션태그로 전환

# 실습: 코드 실행과 분석, 수정하기

