



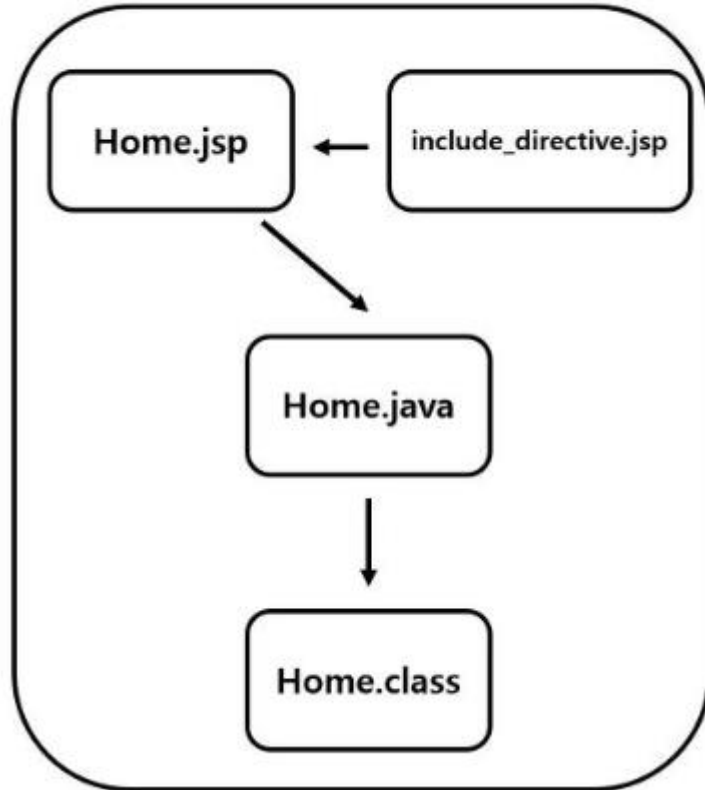
# JSP/Servlet

ch4. JSP 내장객체

1. JSP 내장객체를 이해한다.
2. response 객체를 사용해 페이지를 이동할 수 있다.

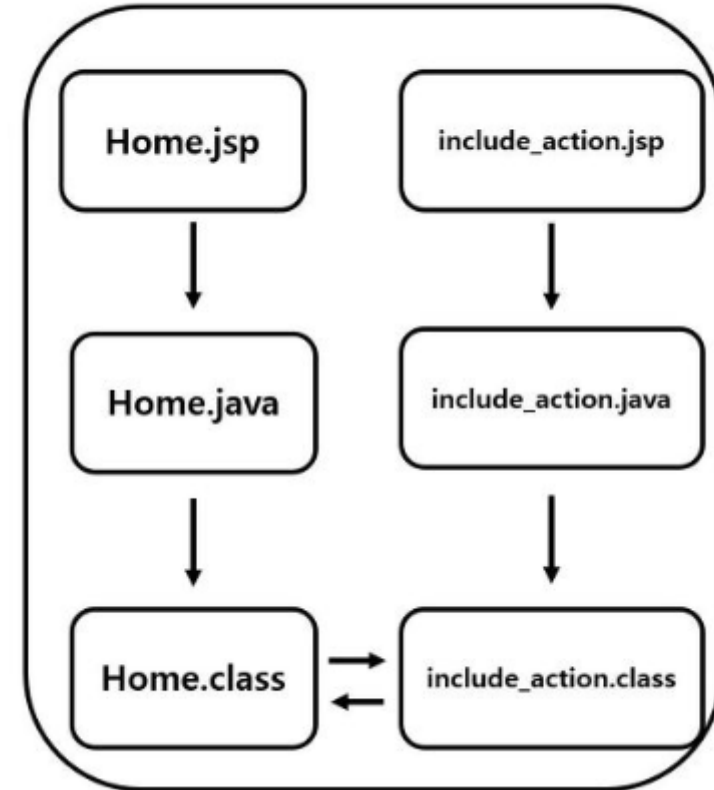
# Include 지시자 vs Include 액션

Include 지시어



`<%@ include file = "jsp"%>`

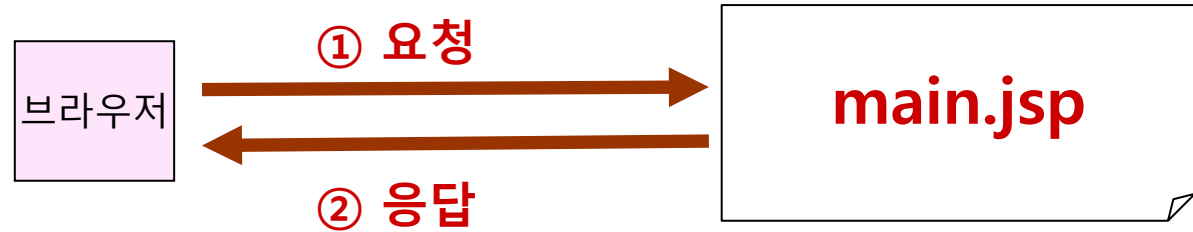
Include 액션



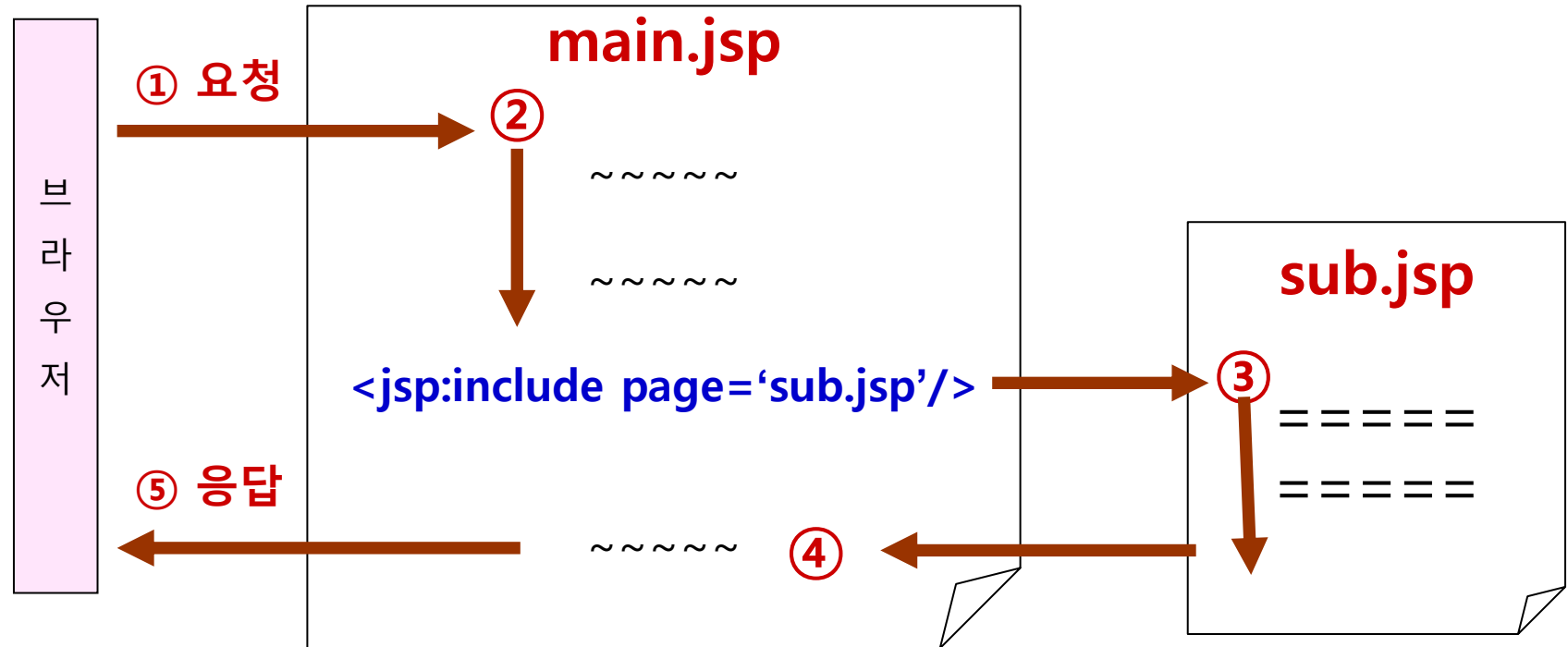
`<jsp:include page = "">`

# Include 지시자 vs Include 액션

**<%page include file>**



**<jsp:include>**



# Include 지시자 vs Include 액션

실습: Include 지시자 파일 코드 수정후 결과 보기

[두번째 페이지로](#)

```
<div style="height:400px;">  
<br><br><br>  
    <a href="mainContent2.jsp">두번째 페이지로</a>  
  
</div>
```

```
<jsp:include page="today.jsp"/>
```

Today: 2022년 8월 10일

## 내장객체(Implicit Object)

```
public void _jspService(final javax.servlet.http.HttpServletRequest request, final
javax.servlet.http.HttpServletResponse response)
    throws java.io.IOException, javax.servlet.ServletException {
```

```
    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;
```

JSP page 안에 선언을 하지 않고도 사용할 수 있는 객체

# 내장객체(Implicit Object)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    int num = Integer.parseInt(request.getParameter("data"));

    for(int i = 1; i <= num; i++){
        out.println(i + "<br>");
    }
%>
```

# 내장객체(Implicit Object)

- JSP page안에서 내장객체를 사용할 수 있는 이유  
Web container가 JSP page를 Servlet class로 변환시키면서 자동으로 내장객체를 선언

변수 이름	제공하는 기능/변수의 역할	변수 타입
request	doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletRequest
response	doGet, doPost 메서드의 두 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletResponse
out	웹 브라우저로 HTML 코드를 출력하는 기능	javax.servlet.jsp.JspWriter
application	JSP 페이지가 속하는 웹 애플리케이션에 관련된 기능	javax.servlet.ServletContext
config	JSP 페이지의 구성 정보를 가져오는 기능	javax.servlet.ServletConfig
pageContext	JSP 페이지 범위 내에서 사용할 수 있는 데이터 저장 기능 등	javax.servlet.jsp.PageContext
Session	세션에 관련된 기능	javax.servlet.http.HttpSession
Page	JSP 페이지로부터 생성된 서블릿	java.lang.Object
Exception	익셉션 객체	java.lang.Throwable



## 내장객체 : out

```
public void _jspService(final javax.servlet.http.HttpServletRequest request, final
javax.servlet.http.HttpServletResponse response)
    throws java.io.IOException, javax.servlet.ServletException {
```

```
    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;
```

Serv  
호출해서 얻는 PrintWriter 객체와 비슷한 역할을

```
        out.write("␣␣␣");
        out.write("<!DOCTYPE html PUBLIC ␣-//W3C//DTD HTML 4.01 Transitional//EN␣"
        "␣␣␣p://www.w3.org/TR/html4/loose.dtd␣␣␣>␣␣␣");
        out.write("<html>␣␣␣");
        out.write("<head>␣␣␣");
        out.write("<meta http-equiv=␣␣Content-Type␣␣ content=␣␣text/html; charset=UTF-8␣␣"
        "␣␣␣");
        out.write("<title>시급계산기</title>␣␣␣");
        out.write("</head>␣␣␣");
        out.write("<body>␣␣␣");
        out.write("␣␣␣<p>␣␣시급 6000원 5시간 계산 : <font color='blue'>");
        out.write("␣␣␣(6000 * 5)");
```

## 내장객체 : out

10개의 방 출력

```
<table border="1">
```

```
<tr>
```

```
<%
```

```
    for(int i=0;i<10;i++){
```

```
        %>
```

```
        <td><%=i%></td>
```

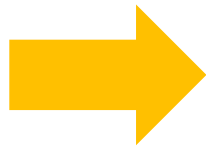
```
        %>
```

```
    }
```

```
%>
```

```
</tr>
```

```
</table>
```



```
<table border="1">
```

```
<tr>
```

```
<%
```

```
    for(int i=0;i<10;i++){
```

```
        out.print("<td>"+i+"</td>");
```

```
    }
```

```
%>
```

```
</tr>
```

```
</table>
```

Scriptlet을 다중으로 사용하지 않음으로  
써 코드가 간결해짐

내장객체 : request

---

# request

사용자 요청과 관련된 기능을 제공하는 내장 객체

(주로 클라이언트에서 서버로 전달되는 정보를 처리하기 위해 사용)

실습6. 정수 1개를 입력받아 입력한 숫자에 따라 방을 생성하시오.

방 이름을 정해주세요

몇 개의 방을 만들까요?

makeTable.jsp

Rose

1	2	3	4	5
---	---	---	---	---


makeTableProcess.jsp

# JSP 실습(필수문제)

실습 1. 아래와 같은 결과를 보이시오.

회원정보

이름	<input type="text" value="홍길동"/>
나이	<input type="text" value="35"/>
성별	<input checked="" type="radio"/> 남자 <input type="radio"/> 여자
좋아하는 음식	<input type="text" value="회덮밥"/> ▼
관심 영역	<input checked="" type="checkbox"/> 웹프로그래밍 <input checked="" type="checkbox"/> 네트워크관리 <input type="checkbox"/> 보안 <input type="checkbox"/> 게임프로그래밍 <input type="checkbox"/> 데이터분석
<input type="button" value="보내기"/>	



사용자 정보

이름	나이	성별	좋아하는 음식	관심분야
홍길동	35(30대)	남자	회덮밥	웹프로그래밍 네트워크관리

# JSP 실습(1, 2조 필수)

**실습 2.** 입력한 데이터들을 아래의 jsp 파일에 전송할 수 있는 조건을 완성하시오.

학점확인프로그램

이름	<input type="text" value="차현석"/>
JAVA점수	<input type="text" value="100"/>
WEB점수	<input type="text" value="100"/>
IOT점수	<input type="text" value="100"/>
ANDROID점수	<input type="text" value="100"/>
<input type="button" value="확인하기!"/>	

scoreForm.html

전송방식 : post

**실습 2.** 입력한 데이터들을 아래의 jsp 파일에 전송할 수 있는 조건을 완성하시오.

학점확인프로그램	
이름	차현석
JAVA점수	100
WEB점수	100
IOT점수	100
ANDROID점수	100
평균	100.0
학점	<b>A+</b> 강조태그적용

scoreResult.jsp

평균과 학점을 구하시오(Average() 메소드 정의)

평균 : 100~95 A+ / 94~90 A / 89~85 B+ / 84~80 B / ~ F

# JSP 과제(3, 4조 필수)

**Step1 : 아이디/비번 입력**

아이디

비밀번호

비밀번호 확인

**Step2 : 개인정보**

성별 : 남자 ☒ 여자 ☐

혈액형 :

생일 :

**Step : 선호도**

취미 : 축구 ☐ 야구 ☒ 농구 ☒

좋아하는 색깔:

**Step4 : 적고 싶은 말**

A02Join.html



id:aaa  
pw:123  
rpw:123  
성별:남자  
blood:B형  
birth:2022-06-30  
야구농구  
color : #f85d5d



message : 안녕하세요

A02Join.jsp



내장객체 : response

---

# response

JSP Page의 실행결과를 웹 브라우저로  
되돌려줄 때 사용하는 내장객체

내장객체 : response

## response.sendRedirect()

현재 실행 중인 JSP page의 실행을 중단하고 다른 Web page가 대신 호출되도록 만드는 기능

```
<form action="neverMove.jsp">  
<input type="submit" value="네이버로 이동">
```

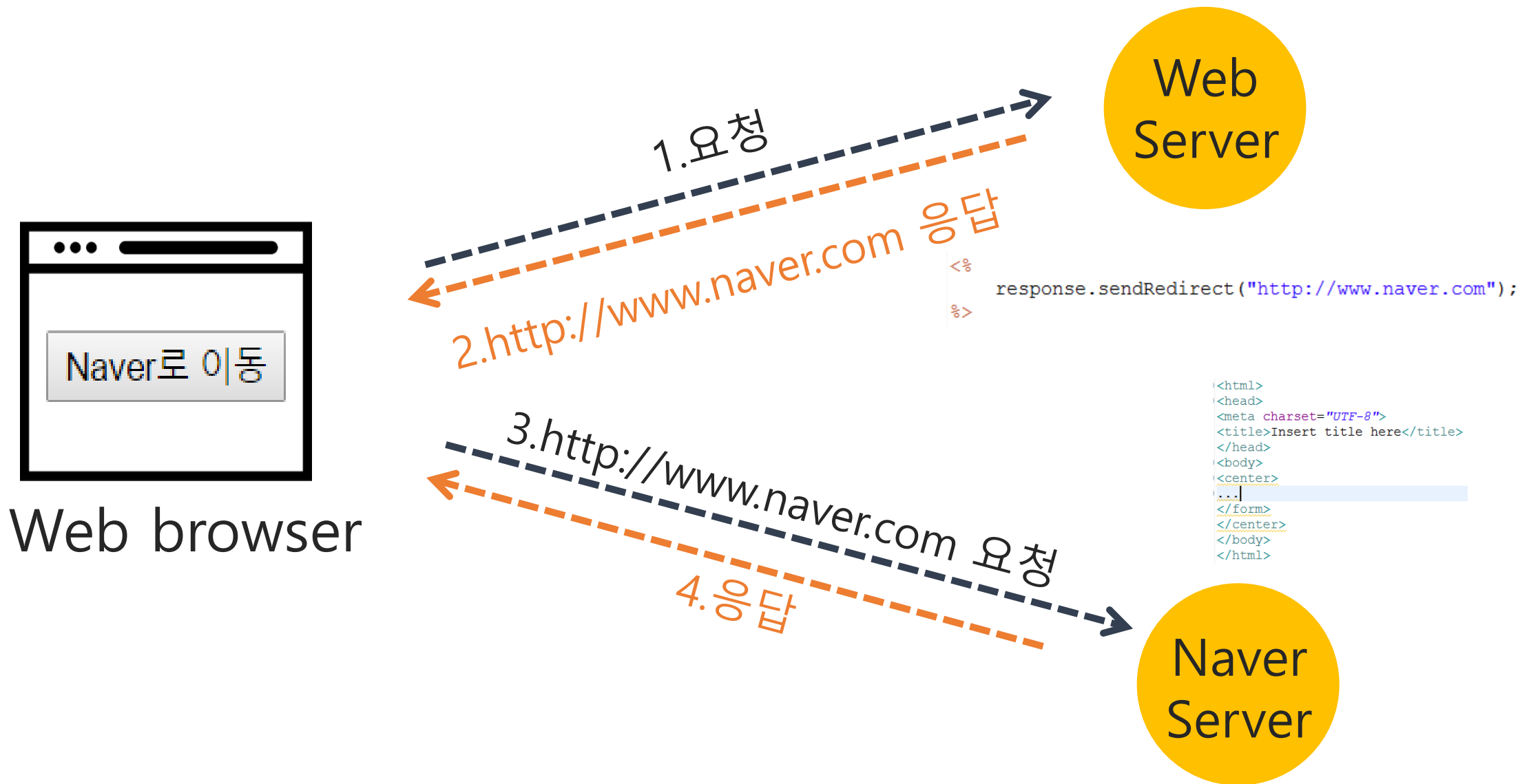
```
<%  
    response.sendRedirect("http://www.naver.com");  
%>
```

naverMove.jsp

이동할 페이지 URL지정



# 내장객체 : response



# JSP 실습

실습 3. 아래와 같은 페이지를 완성하시오.

네이버 ▼

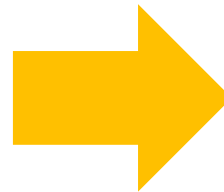
네이버

다음

구글

moveURL.html

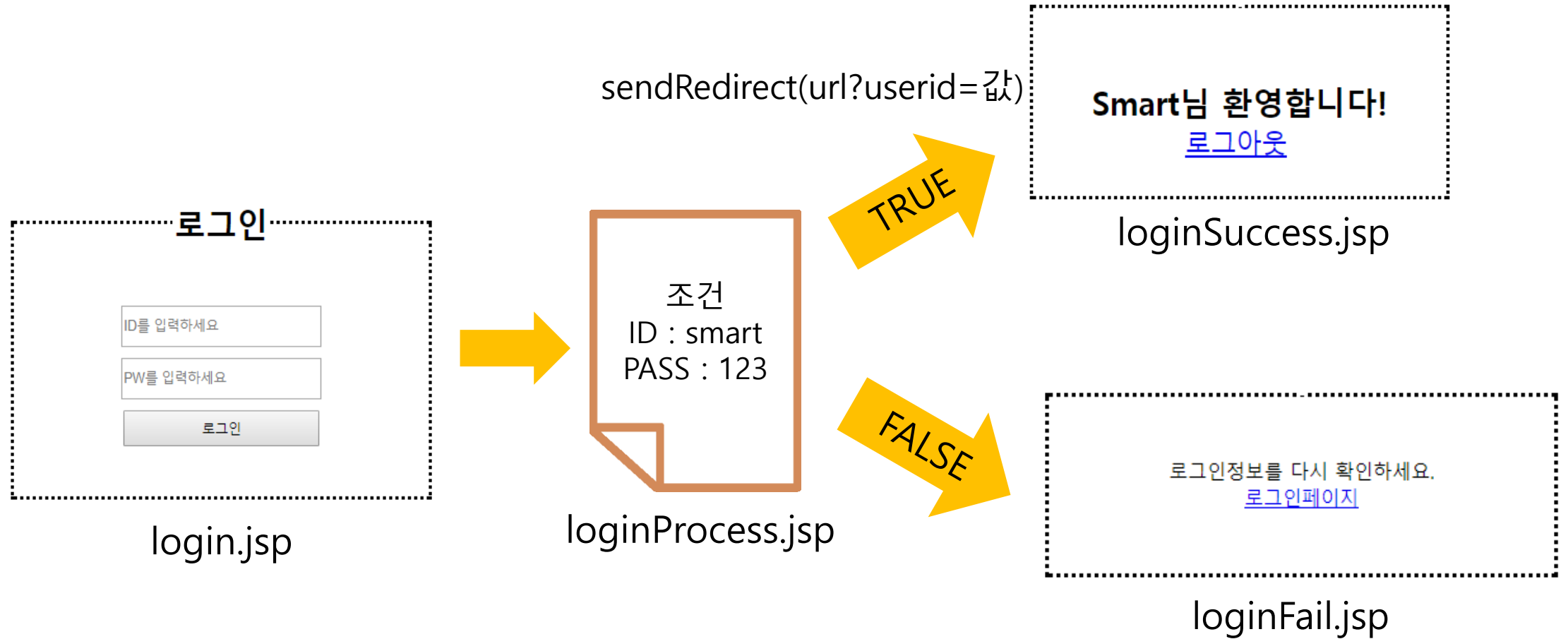
제출



moveURL.jsp

# JSP 실습과제

## 실습 4. 로그인 프로그램을 작성하시오,



```
response.sendRedirect("loginSuccess.jsp?userId="+ URLEncoder.encode(userId, "UTF-8"));
```

한글포함할 경우 인코딩 방식 지정

```
<%  
request.setCharacterEncoding("UTF-8");  
String userId = request.getParameter("userId");  
String pwd = request.getParameter("pwd");  
System.out.println("userId: " + userId);  
System.out.println("pwd: " + pwd);  
if(userId.equals("스마트") && pwd.equals("123")){  
    //request.setAttribute("name", "홍길동");  
  
    RequestDispatcher dispatcher = request.getRequestDispatcher("loginSuccess.jsp");  
    dispatcher.forward(request, response);  
  
    //response.sendRedirect("loginSuccess.jsp?userId=" +  
    //+ URLEncoder.encode(userId, "UTF-8"));  
    //} else response.sendRedirect("loginFail.jsp");  
  
} else response.sendRedirect("loginFail.jsp");  
%>
```

# 웹페이지 이동시키기

## 1. 리다이렉트 방식

1. 사용자 요청이 2번 이루어짐
2. 사용 방법: `response.sendRedirect("url");`

## 2. 포워드 방식:

1. 사용자 요청이 1번 이루어짐
2. 사용자 요청 정보는 request 객체 정보를 통해 공유됨

### 3. 사용 방법:

1. `forward()` 사용을 위한 `RequestDispatcher` 인스턴스 생성

`RequestDispatcher dispatcher = request.getRequestDispatcher("url");`

`dispatcher.forward(request, response);`

2. `request.setAttribute(String, String)`, `request.getAttribute("String")`을 통해 추가적인 값 전달 가능

