

제 08 장

JSP 액션 태그

- 1) 액션 태그의 개요
- 2) 요청흐름제어 <jsp:include> 액션 태그 사용법
- 3) 요청흐름제어 <jsp:forward> 액션 태그 사용법
- 4) param 액션 태그 사용법

액션 태그의 개요



- ❖ 서버나 클라이언트에게 어떤 행동을 하도록 명령하는 태그
- ❖ JSP 페이지에서 페이지와 페이지 사이 제어
- ❖ 다른 페이지의 실행 결과 내용을 현재 페이지에 포함
- ❖ 자바 빈즈(JavaBeans) 등의 다양한 기능 제공
- ❖ XML 형식 <jsp: ... /> 사용



❖ XML 스타일의 태그로 기술

- 특정한 동작 기능을 수행
 - `<jsp:태그키워드 태그속성="태그값" />`
 - `<jsp:include page="sub.jsp" />`

❖ 액션 태그에서 매개변수 지정

- 시작 태그 `<jsp:태그키워드 ... >`와 종료 태그 `</jsp:태그키워드>` 사이에 `<jsp:param ... />`과 같은 param 태그를 기술

```
<jsp:태그키워드 태그속성="태그값">  
    매개변수 지정과 같은 다른 내용  
</jsp:태그키워드>
```

```
<jsp:include page="includesub.jsp" >  
    <jsp:param name="weeks" value="52" />  
</jsp:include>
```

중요 액션 태그



태그 종류	태그 형식	사용 용도
include param	<pre><jsp:include page="test.jsp" /></pre> <pre><jsp:include page="test.jsp" > <jsp:param name="id" value="hong" /> </jsp:include></pre>	현재 JSP 페이지에 다른 페이지를 포함
forward Param	<pre><jsp:forward page="test.jsp" /></pre> <pre><jsp:forward page="test.jsp" > <jsp:param name="id" value="hong" /> </jsp:forward></pre>	현재 JSP 페이지의 제어를 다른 페이지에 전달
plugin	<pre><jsp:plugin type="applet" code="test" /></pre>	자바 애플릿 등을 플러그인
useBean	<pre><jsp:useBean id="login" class="LoginBean" /></pre>	자바 빈즈를 사용
setProperty	<pre><jsp:setProperty name="login" property="pass" /></pre>	자바 빈즈의 속성을 지정하는 메소드를 호출
getProperty	<pre><jsp:getProperty name="login" property="pass" /></pre>	자바 빈즈의 속성을 반환하는 메소드를 호출

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.



- ❖ 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

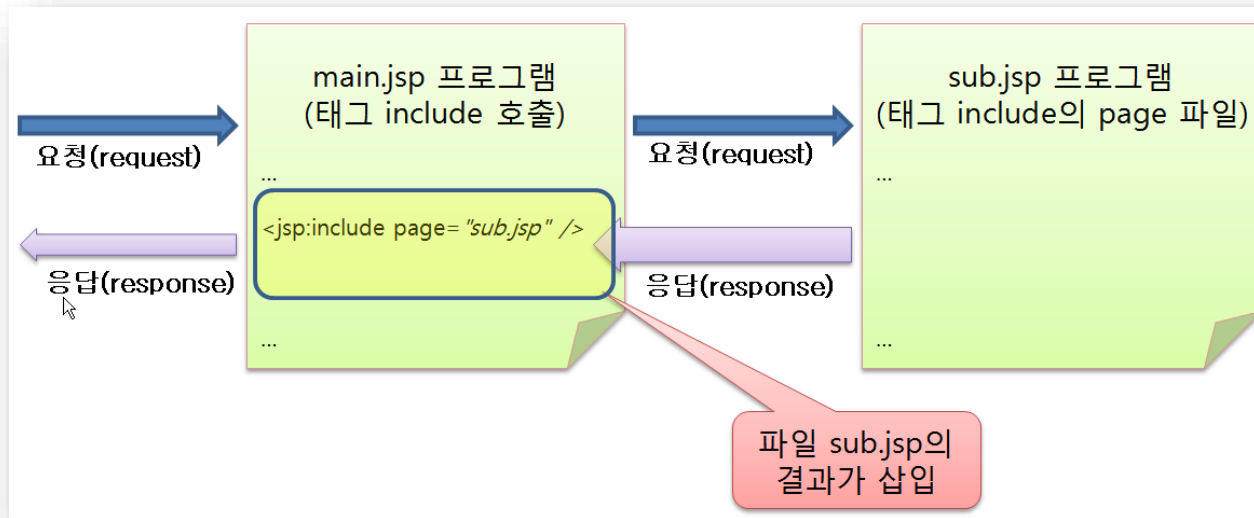
- 현재 JSP 페이지 내에 포함할 내용을 가진 외부 파일명
- 외부 파일은 현재 JSP 페이지와 같은 디렉터리에 있으면 파일명만 설정하고, 그렇지 않으면 전체 URL(또는 상대 경로)을 설정해야 함

❖ flush 속성 값

- 설정한 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장한 결과를 처리
- 기본 값은 false
- true 로 설정하면 외부 파일로 제어가 이동할 때 현재 JSP 페이지가 지금까지 출력 버퍼에 저장된 내용을 웹 브라우저에 출력하고 출력 버퍼를 비움

❖ 속성 page

- 액션 태그 include는 현재의 JSP 페이지에서 기술된 다른 JSP 페이지를 호출하여 그 결과를 include 태그의 위치에 삽입시키는 역할을 수행
- 태그 include에서 속성 page에 삽입할 파일이름을 기술
 - `<jsp:include page="sub.jsp" />`





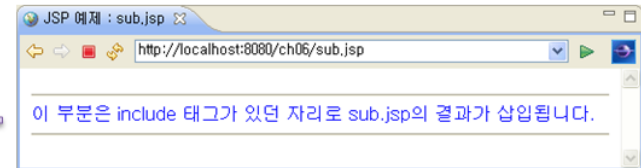
❖ main.jsp, sub.jsp

main.jsp 프로그램

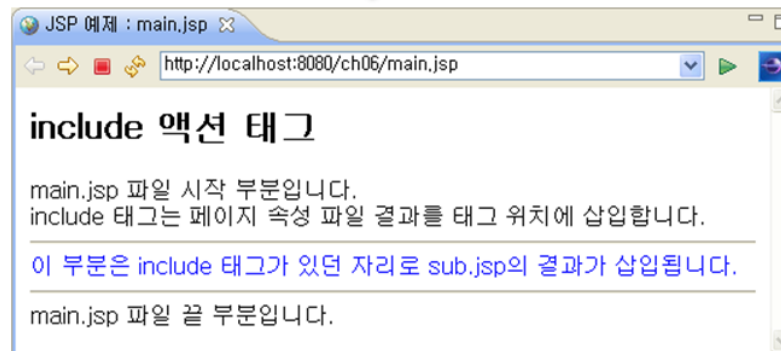
```
<h2> include 액션 태그 </h2>
main.jsp 파일 시작 부분입니다.<br>
include 태그는 페이지 속성 파일 결과를 태그 위치에 삽입합니다.<br>

<jsp:include page="sub.jsp" />

main.jsp 파일 끝 부분입니다.
```



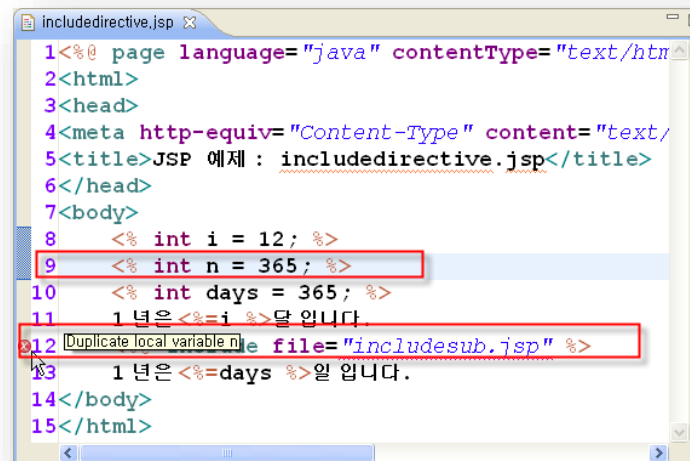
파일 sub.jsp의
결과가 삽입



❖ 소스의 삽입

❖ 변수의 선언이 중복 오류가 발생

- 지시자 `include`가 있는 페이지 `includedirective.jsp`에 변수 `i`와 `n`이 선언되었다고 가정
 - `<% int i = 12; %>`
 - `<% int n = 365; %>`
 - `<%@ include file="includesub.jsp" %>`
- 소스가 삽입되는 페이지 `includesub.jsp`
 - `<% int n = 52; %>`



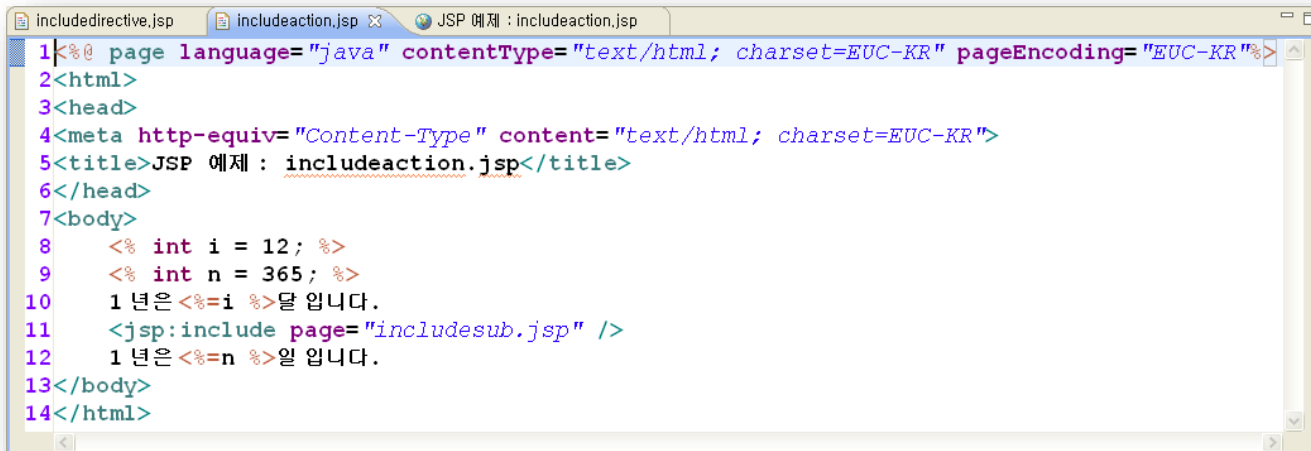
```
1<%@ page language="java" contentType="text/html" %>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html" %>
5<title>JSP 예제 : includedirective.jsp</title>
6</head>
7<body>
8  <% int i = 12; %>
9  <% int n = 365; %>
10  <% int days = 365; %>
11  1 년은 <%=i %> 달 입니다.
12 Duplicate local variable n file="includesub.jsp" %>
13  1 년은 <%=days %> 일 입니다.
14</body>
15</html>
```

❖ 결과의 삽입

- 지시자 `include`와 다르게 액션 태그 `include`를 이용했을 경우는 결과값이 포함되기 때문에 이러한 지역변수 중복 선언의 문제가 발생하지 않음

❖ 액션 태그 `<jsp:include ... />`

- 내장 객체 `pageContext`의 메소드 `include()`와 같은 기능
 - `<% pageContext.include("includesub.jsp"); %>`
 - `<jsp:include page="includesub.jsp" />`



```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 : includeaction.jsp</title>
6</head>
7<body>
8    <% int i = 12; %>
9    <% int n = 365; %>
10    1 년은 <%=i %>달 입니다.
11    <jsp:include page="includesub.jsp" />
12    1 년은 <%=n %>일 입니다.
13</body>
14</html>
```



❖ include 액션 태그와 디렉티브 태그의 차이

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.



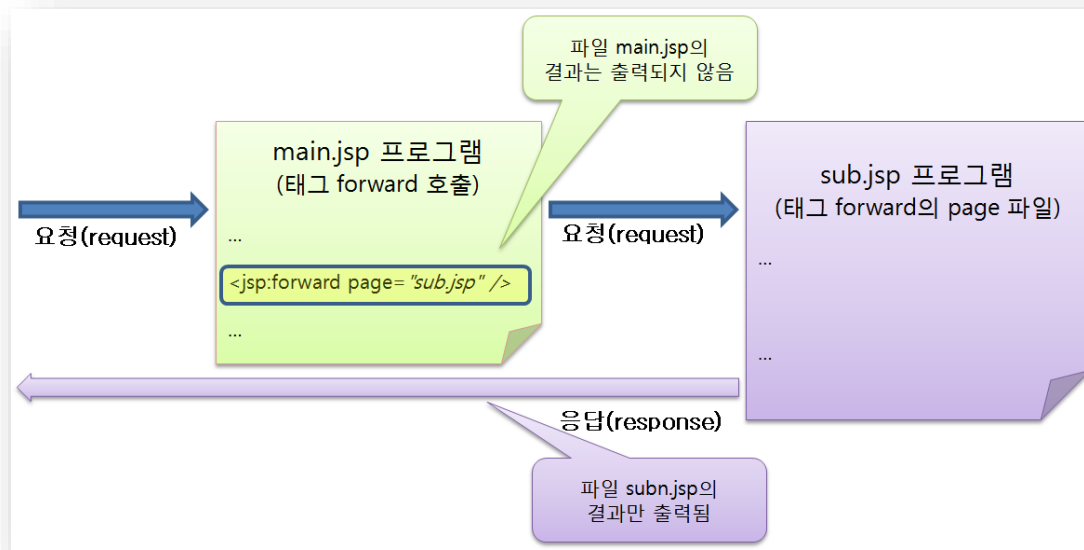
- ❖ 현재 JSP 페이지에서 다른 페이지로 이동하는 태그
- ❖ JSP 컨테이너는 현재 JSP 페이지에서 forward 액션 태그를 만나면
 - 그 전까지 출력 버퍼에 저장되어 있던 내용을 모두 삭제하고
 - forward 액션 태그에 설정된 페이지로 프로그램의 제어가 이동

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

- ❖ 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 설정

❖ 속성 page

- 속성 page에 지정한 JSP 페이지 또는 파일을 호출하는 기능
 - `<jsp:forward page="forwardsub.jsp" />`
- forward 태그가 있는 현재 페이지의 작업은 모두 중지 되고,
- 이전에 출력한 버퍼링 내용도 모두 사라지게 되어 출력이 되지 않으며
- 모든 제어가 page에 지정한 파일로 이동



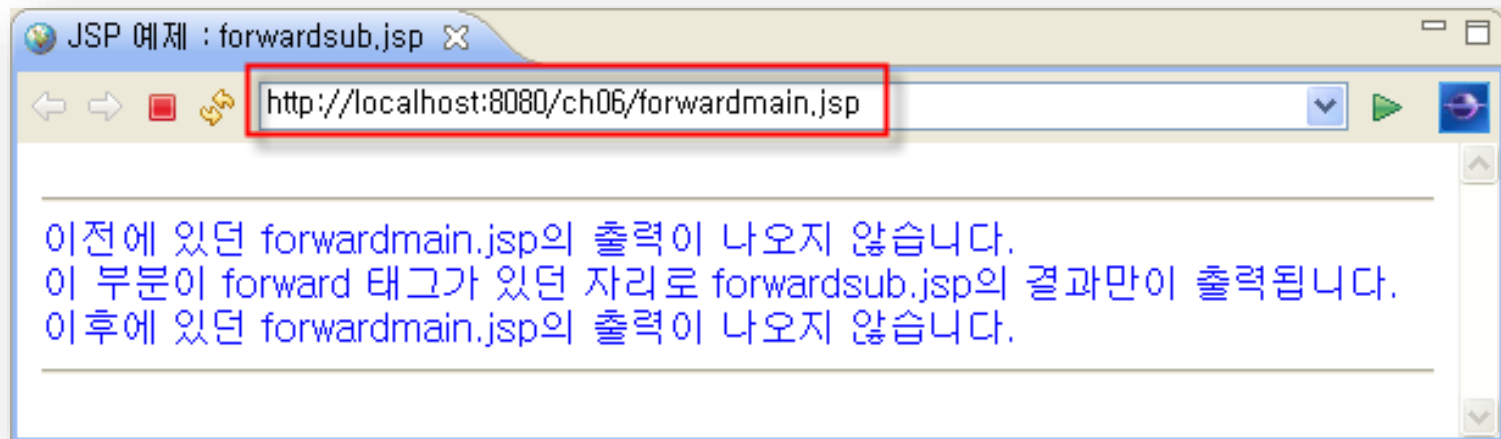
태그 *forward*와 *include*의 차이

❖ 태그 *include*

- *page* 속성에 지정된 페이지의 처리가 끝나면 다시 현재 페이지로 돌아와 처리를 진행

❖ 태그 *forward*

- *page* 속성에 지정된 페이지로 제어가 넘어가면 다시 현재 페이지로 다시 돌아오지 않고 이동된 페이지에서 실행을 종료



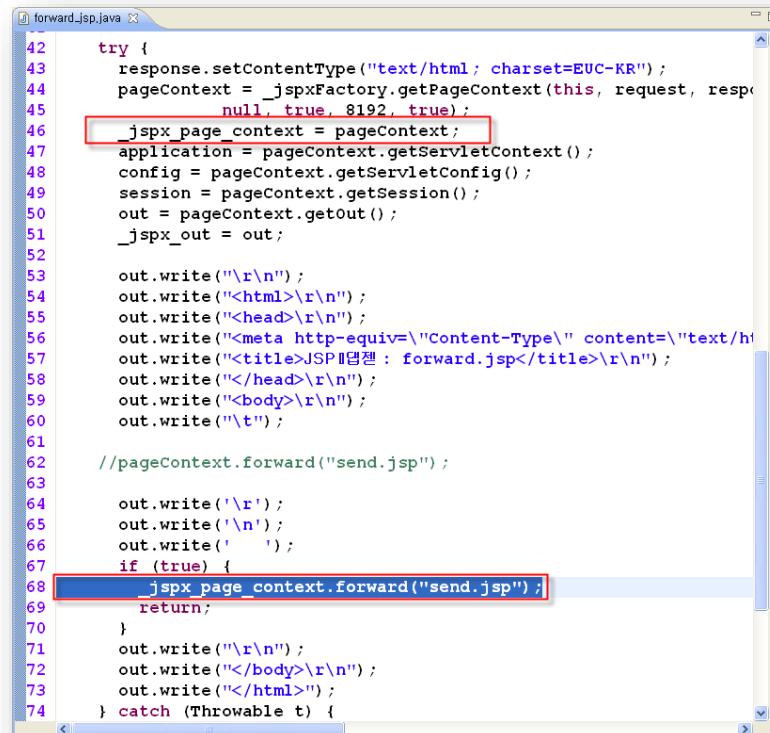
pageContext.forward()

❖ 액션 태그 forward

- 실제 JSP 서블릿 소스에서 내장 객체 pageContext의 메소드 forward()로 대체
- pageContext.forward() 같은 기능을 수행

```
<%  
    pageContext.forward("send.jsp");  
%>
```

```
<jsp:forward page="send.jsp" />
```



```
42     try {  
43         response.setContentType("text/html; charset=EUC-KR");  
44         pageContext = _jspxFactory.getPageContext(this, request, response, null, true, 8192, true);  
45         _jspx_page_context = pageContext;  
46         application = pageContext.getServletContext();  
47         config = pageContext.getServletConfig();  
48         session = pageContext.getSession();  
49         out = pageContext.getOut();  
50         _jspx_out = out;  
51  
52         out.write("\r\n");  
53         out.write("<html>\r\n");  
54         out.write("<head>\r\n");  
55         out.write("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=EUC-KR\">\r\n");  
56         out.write("<title>JSP예제 : forward.jsp</title>\r\n");  
57         out.write("</head>\r\n");  
58         out.write("<body>\r\n");  
59         out.write("\t");  
60  
61         //pageContext.forward("send.jsp");  
62  
63         out.write("\r");  
64         out.write("\n");  
65         out.write(" ");  
66         if (true) {  
67             _jspx_page_context.forward("send.jsp");  
68             return;  
69         }  
70         out.write("\r\n");  
71         out.write("</body>\r\n");  
72         out.write("</html>");  
73     } catch (Throwable t) {  
74     }
```



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

```
<jsp:forward page="파일명">  
    <jsp:param name="변수명1" value="변수값1" />  
    <jsp:param name="변수명2" value="변수값2" />  
</jsp:forward>
```




❖ 태그 param

- 태그 <jsp:include ... >와 <jsp:forward ... >와 함께 사용
- page에 지정된 페이지로 필요한 파라미터의 이름(name)과 값(value)을 전송하는 역할을 수행
- 태그 param은 속성 name과 value를 제공

```
<jsp:include page="/loginhandle.jsp" >  
    <jsp:param name="userid" value="guest" />  
    <jsp:param name="passwd" value="anonymous" />  
</jsp:include>
```

❖ 태그 *include*에서 지정한 인자의 전송

```
<jsp:include page="loginhandle.jsp" >  
    <jsp:param name="userid" value="guest" />  
    <jsp:param name="passwd" value="anonymous" />  
</jsp:include>
```

```
<%  
    if (userid.equals("")) {  
        %>  
            <jsp:include page="loginhandle.jsp" >  
                <jsp:param name="userid" value="guest" />  
                <jsp:param name="passwd" value="anonymous" />  
            </jsp:include>  
        <%  
            } else {  
                %>  
                    <jsp:include page="loginhandle.jsp" />  
                <%  
            }  
        %>
```

❖ 태그 forward에서 지정한 인자의 전송

```
<jsp:forward page="forwardloginhandle.jsp" >  
    <jsp:param name="snum" value="1234-5678" />  
</jsp:forward>
```

```
<%  
if ( userid == null && passwd == null ) {  
%>  
    <jsp:forward page="forwardloginhandle.jsp" />  
  
<%  
} else {  
%>  
    <jsp:forward page="forwardloginhandle.jsp" >  
        <jsp:param name="snum" value="1234-3459" />  
    </jsp:forward>  
  
<%  
}  
%>
```



❖ plugin

- 웹브라우저에서 자바 빈즈 또는 애플릿을 플러그인하여 실행하는 태그
- 태그 **plugin**은 각기 다른 웹 브라우저에서 인식할 수 있도록 마이크로소프트사의 IE 경우일 때는 **OBJECT** 태그로 만들어 주며, 넷스케이프 사의 경우 **EMBED** 형태의 태그로 만들어 줌

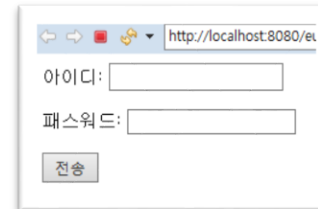
```
<jsp:plugin
  type = "bean | applet"
  code = "objectCode"
  codebase="objectCodebase"
  align="alignment"
  width = "width"
  nspluginurl = "url"
  iepluginurl = "url" >
  <jsp:params name="paramName" value="paramValue" />
  <jsp:fallback> arbitrary_text </jsp:fallback> >
</jsp:plugin>
```

[실습1]



```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="EUC-KR">
7   <title>Insert title here</title>
8 </head>
9 <body>
10
11 <form name="test" method=post action="reqfrom.jsp">
12 아이디: <input type=text name=id size=20>
13 <p>
14 비밀번호: <input type=password name=pw size=20>
15 <p>
16 <input type=submit value=전송 ><br>
17
18 </form>
19 </body>
20 </html>
```

// main.jsp

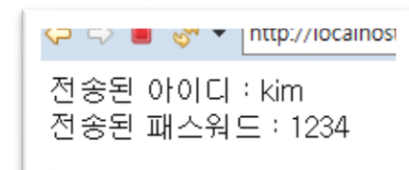


```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4   <jsp:forward page="reqto.jsp" />
```

// reqfrom.jsp

```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4 <%
5   request.setCharacterEncoding("euc-kr");
6
7   String nid = request.getParameter("id");
8   String npw = request.getParameter("pw");
9
10  out.println("전송된 아이디 : " + nid + "<br>");
11  out.println("전송된 패스워드 : " + npw + "<br>");
12
13 %>
14
```

// reqto.jsp





```

1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="EUC-KR">
7   <title>Insert title here</title>
8 </head>
9 <body>
10
11 <form name="test" method="post" action="reqfrom2.jsp">
12   아이디: <input type="text" name="id" size="20">
13   <p>
14   비밀번호: <input type="password" name="pw" size="20">
15   <p>
16   이름: <input type="text" name="uname" size="20">
17   <p>
18   <input type="submit" value="전송" ><br>
19
20 </form>
21 </body>

```

// main2.jsp

← → ↻ ↺ http://localhost:

전송된 아이디 : kim
 전송된 비밀번호 : 1234
 전송된 아이디 : 홍길동

```

1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4
5 <%
6   request.setCharacterEncoding("euc-kr");
7
8   String nid = request.getParameter("id");
9   out.println("전송된 아이디 : " + nid + "<br>");
10
11 %>
12
13 <jsp:include page="reqto2.jsp" flush="false" />
14
15 <%
16
17   String nuname = request.getParameter("uname");
18   out.println("전송된 아이디 : " + nuname + "<br>");
19
20 %>

```

// reqfrom2.jsp

```

1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4 <%
5   request.setCharacterEncoding("euc-kr");
6
7   String npw = request.getParameter("pw");
8   out.println("전송된 비밀번호 : " + npw + "<br>");
9
10 %>

```

// reqto2.jsp



```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4                                     // main3.jsp
5 <%
6   request.setCharacterEncoding("euc-kr");
7
8   out.println("첫번째 화면입니다." + "<br><hr>");
9 %>
10
11 <jsp:include page="paramto.jsp" flush="true">
12   <jsp:param name="ko" value="한국" />
13   <jsp:param name="am" value="미국" />
14 </jsp:include>
15
16
17
18
19 <%
20   out.println("<hr> 마지막 화면입니다." + "<br>");
21 %>
```

← → ⏏ 🔄 http://localhost:8080/eu

첫번째 화면입니다.

문서의 처리 결과입니다.
전송된 ko는 한국입니다.
전송된 am은 미국입니다.

마지막 화면입니다.

```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3
4
5 <%
6   request.setCharacterEncoding("euc-kr");
7
8   String nko = request.getParameter("ko");
9   String nam = request.getParameter("am");
10  out.println("문서의 처리 결과입니다.<br>");
11
12
13 out.println("전송된 ko는 " + nko + "입니다.<br>");
14 out.println("전송된 am은 " + nam + "입니다.<br>");
15
16 %>
```

// paramto.jsp



- ❖ 강의 동영상 자료에 있는 실습 문제(총 3개)를 모두 프로그램 한 후 소스와 출력 결과를 편집하여 제출하세요.

-
- ❖ 제출방법 : ppt파일에 소스와 출력 결과 편집 후 제출
 - ❖ 제출마감일 : 2021년 05월03일 월요일 23시 55분까지
 - ❖ 제출장소 : 한림 스마트 LEAD(<https://smartlead.hallym.ac.kr>) 해당 과목 [과제]란에 제출하시면 됩니다. (제출시 제목란에 “여러분의 학번이름”을 쓰시고, 파일을 전송하시면 됩니다.)