

JSP 프로그래밍

(재)대덕인재개발원

07 페이지 모듈화와 요청 흐름 제어

- ▶ 1. <jsp:include>액션 태그를 이용한 페이지 모듈화
- ▶ 2. include 디렉티브를 이용한 중복된 코드 삽입
- ▶ 3. <jsp:forward>액션 태그를 이용한 JSP 페이지 이동
- ▶ 4. <jsp:include>/ <jsp:forward> 액션 태그 page 속성의 경로
- ▶ 5. 기본 객체의 속성을 이용해서 값 전달하기



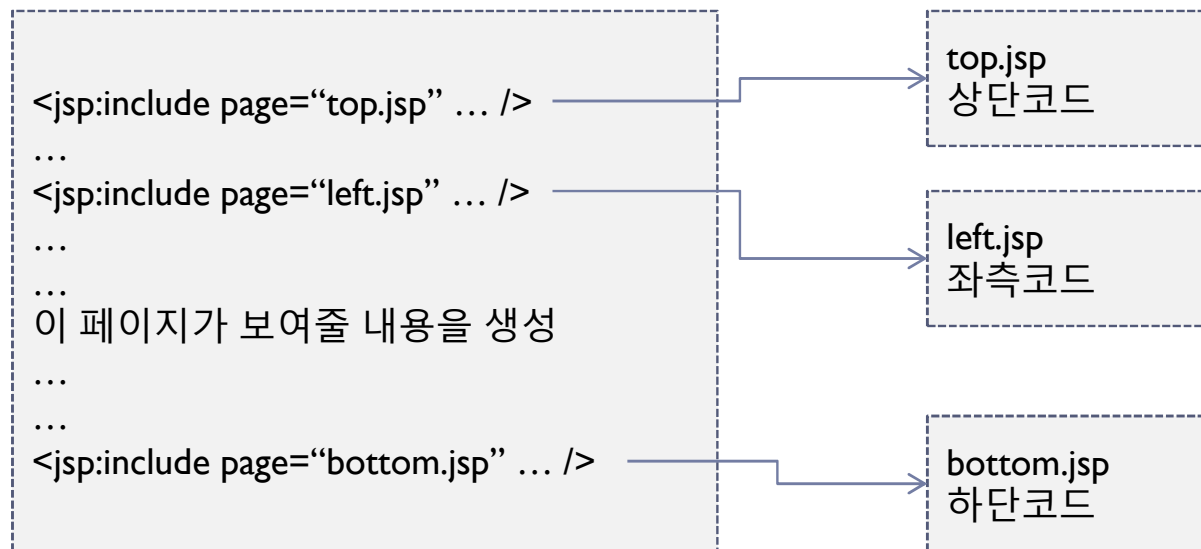
7.1 <jsp:include>액션 태그를 이용한 페이지 모듈화

- ▶ <jsp:include> 액션 태그는 지정한 페이지를 태그가 위치한 부분에 포함시킬 때 사용된다.
- ▶ <jsp:include> 액션 태그는 포함할 JSP 페이지의 실행 결과를 현재 위치에 포함시킨다고 할 수 있다.
- ▶ <jsp:include page="포함할 페이지" flush="true">



7.1.2 <jsp:include> 액션 태그를 이용한 중복 영역의 처리

- ▶ 공통된 코드를 모든 JSP 페이지가 포함하고 있을 경우, 공통 부분이 변경되면, 공통된 부분을 포함하고 있던 모든 JSP 페이지를 변경해 주어야 하는 문제가 발생.
- ▶ <jsp:include> 액션 태그를 사용하면 공통되는 부분의 수정에 따른 문제를 최소화 할 수 있다.



[그림] 공통부분을 별도의 JSP 페이지로 작성한 후 <jsp:include>로 포함시킨다.

7.1.3 <jsp:param>을 이용해서 포함될 페이지에 파라미터 추가하기

- ▶ <jsp:include> 액션 태그는 <jsp:param> 액션 태그를 이용해서 포함할 JSP 페이지에 패러미터를 추가할 수 있다.
- ▶ <jsp:param> 액션 태그는 <jsp:include> 액션 태그의 자식 태그로 추가되며, <jsp:param> 액션 태그의 name 속성과 value 속성에는 추가할 패러미터의 이름과 값을 입력한다.
- ▶ <jsp:param> 액션 태그는 이미 동일한 이름의 패러미터가 존재할 경우, 기존 패러미터 값을 유지하면서 새로운 값을 추가한다.
- ▶ <jsp:param> 액션 태그를 통해서 추가되는 패러미터는 <jsp:include>에 포함되는 페이지에서만 유효하다

```
<jsp:include page="/module/top.jsp" flush="false" >
    <jsp:param name="name" value="이산" /> <!-- value에 값을 직접 입력-->
    <jsp:param name="type" value="<%= type %>" /> <!-- 표현식으로 값 입력-->
</jsp:include>
```

7.2 include 디렉티브를 이용한 중복된 코드 삽입

- ▶ include 디렉티브도 <jsp:include>와 마찬가지로 지정한 페이지를 현재 위치에 포함시켜 주는 기능을 제공한다.
- ▶ <jsp:include>는 다른 JSP로 흐름을 이동시켜 그 결과물을 현재 위치에 포함시키는 방식인 반면에, include 디렉티브는 다른 파일의 내용을 현재 위치에 삽입한 후에 JSP 파일을 자바 파일로 변환하고 컴파일 하는 방식이다.
- ▶ <%@ include file="포함할 파일"%>
- ▶ include 디렉티브는 코드 차원에서 삽입이 이루어지기 때문에 삽입되는 파일에서 선언한 변수를 삽입하는 JSP에서 사용할 수 있다.
- ▶ 모든 JSP 페이지에서 사용되는 변수 지정 및 저작권 표시와 같은 간단하면서도 모든 페이지에서 중복되는 문장 등에 활용할 수 있다.



7.2.1 include 디렉티브의 처리 방식과 사용법


```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head><title>include 디렉티브</title></head>
<body>

<%
    int number = 10;
%>

<%@ include file="/includee.jspf" %>
공통 변수 DATAFOLDER = "<%= dataFolder %>"

</body>
</html>
```

```
<%@ page contentType="text/html; charset=utf-8" %>
includer.jsp에서 지정한 번호 : <%= number %>
<p>
<%
    String dataFolder = "C:/data";
%>
```



7.2.3 코드 조각 자동 포함 기능(JSP 2.0)

- ▶ JSP 2.0 버전부터 include 디렉티브를 사용하지 않고도 JSP의 앞뒤에 지정한 파일을 삽입하는 기능을 제공하고 있다.

```
<%@ page contentType="text/html; charset=utf-8" %>
<%@ include file="/common/variable.jspf" %>
<html>
...
...
<%@ include file="/common/footer.jspf" %>
```

```
<jsp-config>
  <jsp-property-group>
    <url-pattern>/view/*</url-pattern>
    <include-prelude>/common/variable.jspf</include-prelude>
    <include-coda>/common/footer.jspf</include-coda>
  </jsp-property-group>
</jsp-config>
```

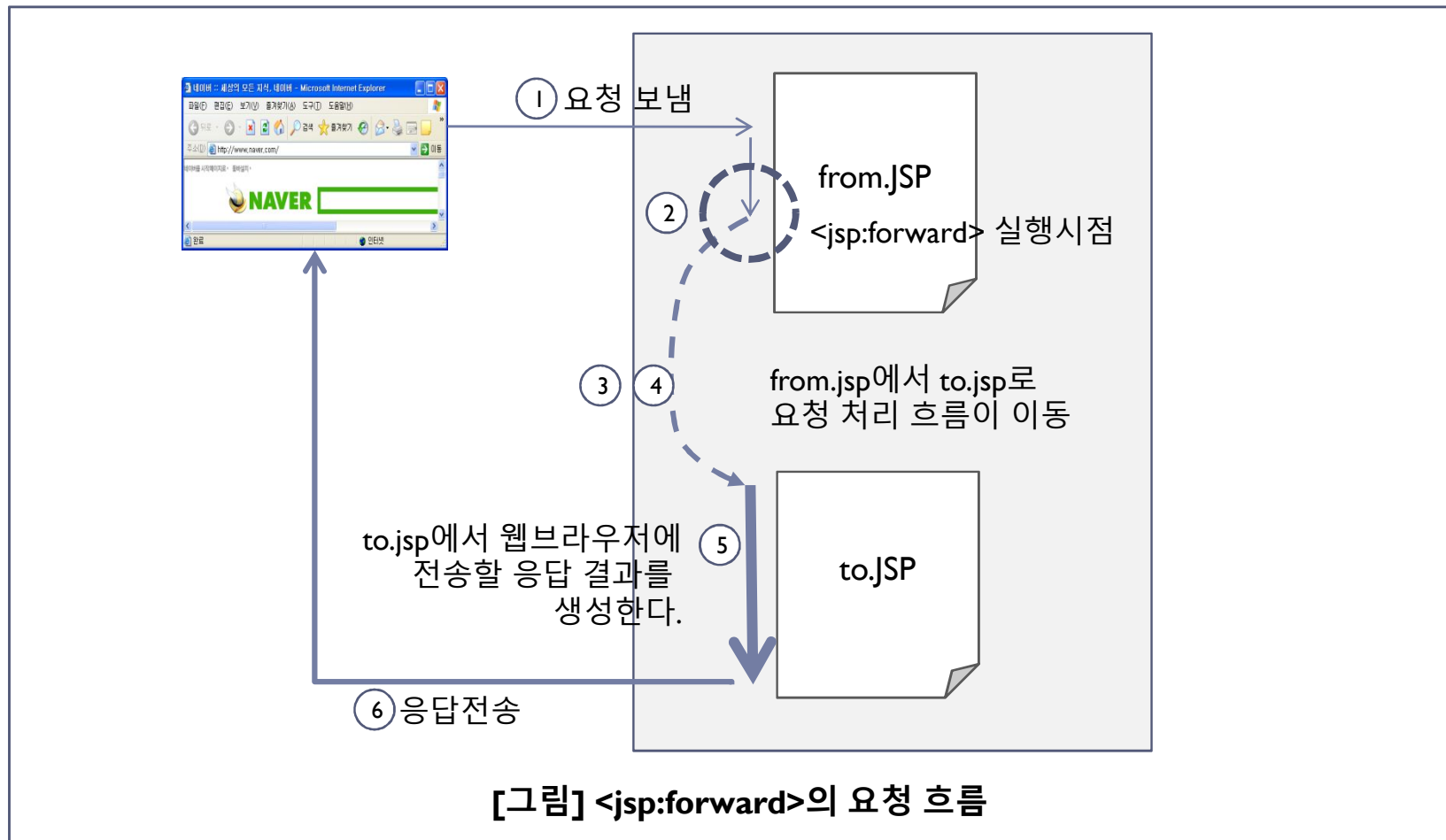

7.2.4 <jsp:include>액션 태그와 include 디렉티브의 비교

비교 항목	<jsp:include>	include 디렉티브
처리 시간	요청 시간에 처리	JSP 파일을 자바 소스로 변환할 때 처리
기능	별도의 파일로 요청 처리 흐름을 이동	현재 파일에 삽입시킴
데이터 전달방법	request 객체나 <jsp:param>을 이용한 패러미터 전달	페이지 내의 변수를 선언한 후, 변수에 값 저장
용도	화면의 레이아웃의 일부분을 모듈화 할 때 주로 사용된다.	다수의 JSP 페이지에서 공통으로 사용되는 변수를 지정하는 코드나 저작권과 같은 문장을 포함한다.



7.3 <jsp:forward>액션 태그를 이용한 JSP 페이지 이동

- ▶ <jsp:forward>액션 태그는 하나의 JSP 페이지에서 다른 JSP 페이지로 요청 처리를 전달할 때 사용된다.



7.3 <jsp:forward>액션 태그를 이용한 JSP 페이지 이동

- ▶ <jsp:forward> 요청 흐름
 - ▶ 1. 웹 브라우저의 요청이 from.jsp에 전달된다.
 - ▶ 2. from.jsp는 <jsp:forward> 액션 태그를 실행한다.
 - ▶ 3. <jsp:forward> 액션 태그가 실행되면 요청 흐름이 to.jsp로 이동한다.
 - ▶ 4. 요청 흐름이 이동할 때 from.jsp에서 사용한 request 객체와 response 객체가 to.jsp로 전달된다.
 - ▶ 5. to.jsp는 응답 결과를 생성한다.
 - ▶ 6. to.jsp가 생성한 결과가 웹 브라우저에 전달된다.

- ▶ 잊지 말아요!
 - ▶ from.jsp가 아닌 to.jsp가 생성한 응답 결과가 웹 브라우저에 전달된다.
 - ▶ from.jsp에서 사용한 request, response 객체가 to.jsp에 그대로 전달된다.



7.3.1 <jsp:forward>액션 태그의 사용법

▶ <jsp:forward page="이동할 페이지" />

```
#fileName = from.jsp
```

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

```
<%--
```

<jsp:forward> 액션 태그를 실행하면 생성했던
출력 결과는 모두 제거된다.

```
--%>
```

```
<html>
```

```
<head><title>from.jsp의 제목</title></head>
```

```
<body>
```

이 페이지는 from.jsp가 생성한 것입니다.

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

```
</body>
```

```
</html>
```

```
#fileName = to.jsp
```

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

```
<html>
```

```
<head><title>to.jsp의 실행 결과</title></head>
```

```
<body>
```

이 페이지는 to.jsp가 생성한 것입니다.

```
</body>
```

```
</html>
```

7.3.2 <jsp:forward>액션 태그와 출력 버퍼와의 관계

- ▶ 교재 P43 참고



7.3.3 <jsp:forward>액션 태그의 전형적인 사용법

- ▶ <jsp:forward>액션 태그를 사용해서 조건에 따라 알맞은 페이지로 분기하도록 하는 것이 <jsp:forward>액션 태그의 일반적인 사용법이다.

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    String forwardPage = null;
    // 조건에 따라 이동할 페이지를 지정
    if(option.equals("A")){
        forwardPage = "/view/typeA.jsp";

    }else if(option.equals("B")){
        forwardPage = "/view/typeB.jsp";

    } else if(option.equals("C")){
        forwardPage = "/view/typeC.jsp";
    }

%>

<jsp:forward page="<%= forwardPage %>" />
```

7.3.4 <jsp:forward>액션 태그를 이용해서 이동할 페이지에 패러미터 추가하기

- ▶ <jsp:forward>액션 태그를 사용해서 이동할 페이지에 추가적으로 정보를 전달하고 싶은 경우에 <jsp:param> 태그를 사용하면 된다.

```
<jsp:forward page="<%= forwardPage %>" >  
    <jsp:param name = "first" value = "SAN" />  
    <jsp:param name = "last" value = "LEE" />  
</jsp:forward>
```



7.4 <jsp:include> / <jsp:forward>액션 태그 page 속성의 경로

- ▶ <jsp:include>액션 태그와 <jsp:forward>액션 태그는 page 속성을 사용해서 포함 혹은 이동할 페이지의 경로를 입력한다.
 - ▶ 웹 어플리케이션 디렉토리를 기준으로 한 절대 경로 :
 - ▶ <jsp:forward page="/to/to.jsp" />
 - ▶ 현재 JSP페이지를 기준으로 한 상대 경로 :
 - ▶ <jsp:forward page="../to/to.jsp" />



7.5 기본 객체의 속성을 이용해서 값 전달하기

- ▶ `<jsp:include>`액션 태그와 `<jsp:forward>`액션 태그는 `<jsp:param>`태그를 이용해서 패러미터를 추가로 전달할 수 있다.
- ▶ 하지만 패러미터는 String 타입의 값만 전달 할 수 있는 제약을 갖는다.
- ▶ 전달할 데이터가 String 타입이 아니면 기본 객체의 속성을 이용해서 전달하는 것이 편리하다.
- ▶ 포함하거나 이동할 페이지는 동일한 요청 범위(request범위)를 갖기 때문에 일반적으로 request 기본 객체의 속성을 이용해서 필요한 값을 전달한다.



7.5 기본 객체의 속성을 이용해서 값 전달하기

```
<%@ page contentType="text/html; charset=utf-8" %>
<%@ page import="java.util.Calendar" %>
<%
    Calendar cal = Calendar.getInstance();
    request.setAttribute("time", cal);
%>
<jsp:forward page="/to/viewTime.jsp" />
```

```
<%@ page contentType="text/html; charset=utf-8" %>
<%@ page import="java.util.Calendar" %>
<html>
<head><title> 현재 시간 </title></head>
<body>
<%
    Calendar cal = (Calendar)request.getAttribute("time");
%>
현재 시간은 <%= cal.get(Calendar.HOUR) %>시
             <%= cal.get(Calendar.MINUTE) %>분
             <%= cal.get(Calendar.SECOND) %>초 입니다.

</body>
</html>
```