

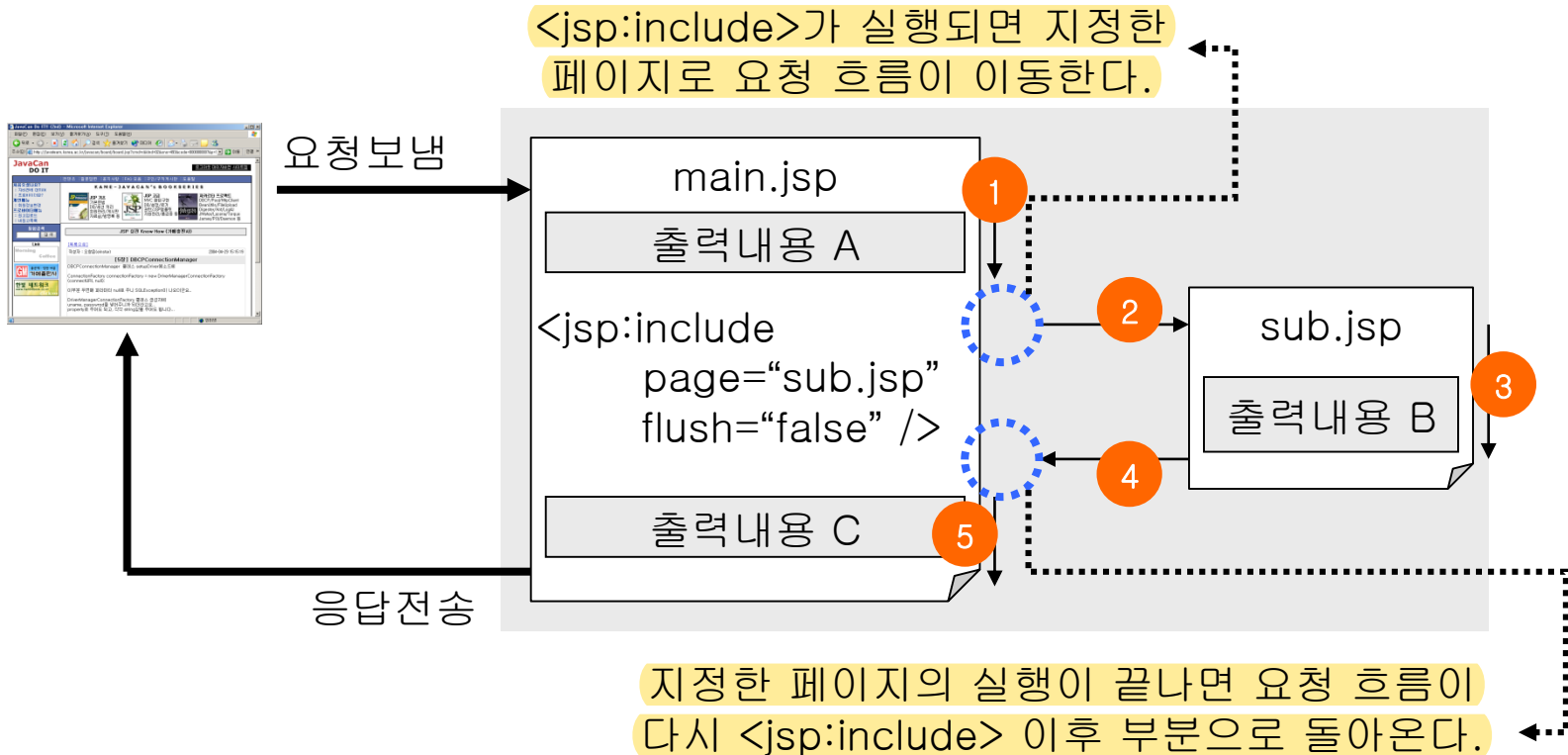
JSP 페이지 모듈화

1. <jsp:include> 액션 태그와 페이지 모듈화 파일 이동 -> 실행 -> 원래 파일
2. include 디렉티브와 페이지 모듈화 파일 내용 복붙 -> 실행
3. <jsp:include>와 include 디렉티브의 비교

<jsp:include> 액션 태그

"<jsp:include> 액션 태그는 지정한 페이지를 태그가 위치한 부분에 포함시킬 때 사용"

동작방식



<jsp:include> 액션 태그

Syntax

```
<jsp:include page="이동할 페이지" flush="true"/>
```

```
<jsp:include page="이동할 페이지" flush="false"/>
```

또는

```
<jsp:include page="이동할 페이지" flush="...">
```

```
    <jsp:param name="파라미터1" value="값1" />
```

```
    <jsp:param name="파라미터2" value="값2" />
```

```
</jsp:include>
```

<jsp:include> 액션 태그 예

Example

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<html>
<head><title>main</title></head>
<body>
```

main.jsp에서 생성한 내용.

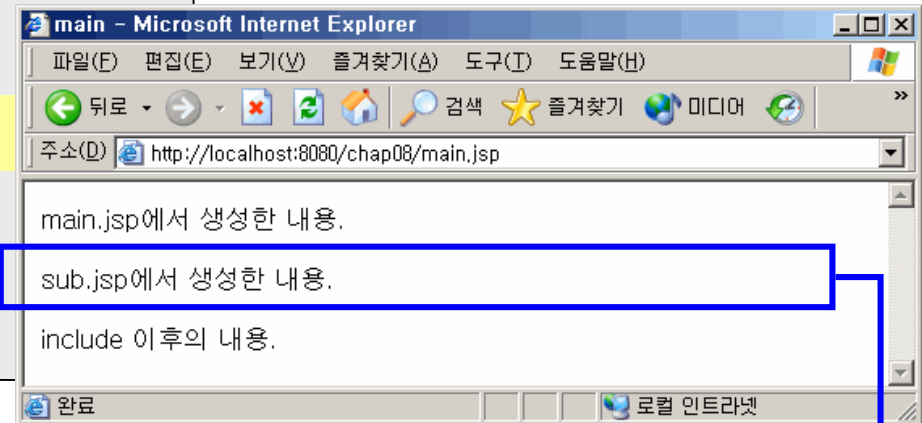
```
<jsp:include page="sub.jsp" flush="false" />
```

include 이후의 내용.

```
</body>
</html>
```

```
<%@ page contentType = "text/html; charset=euc-kr" %>

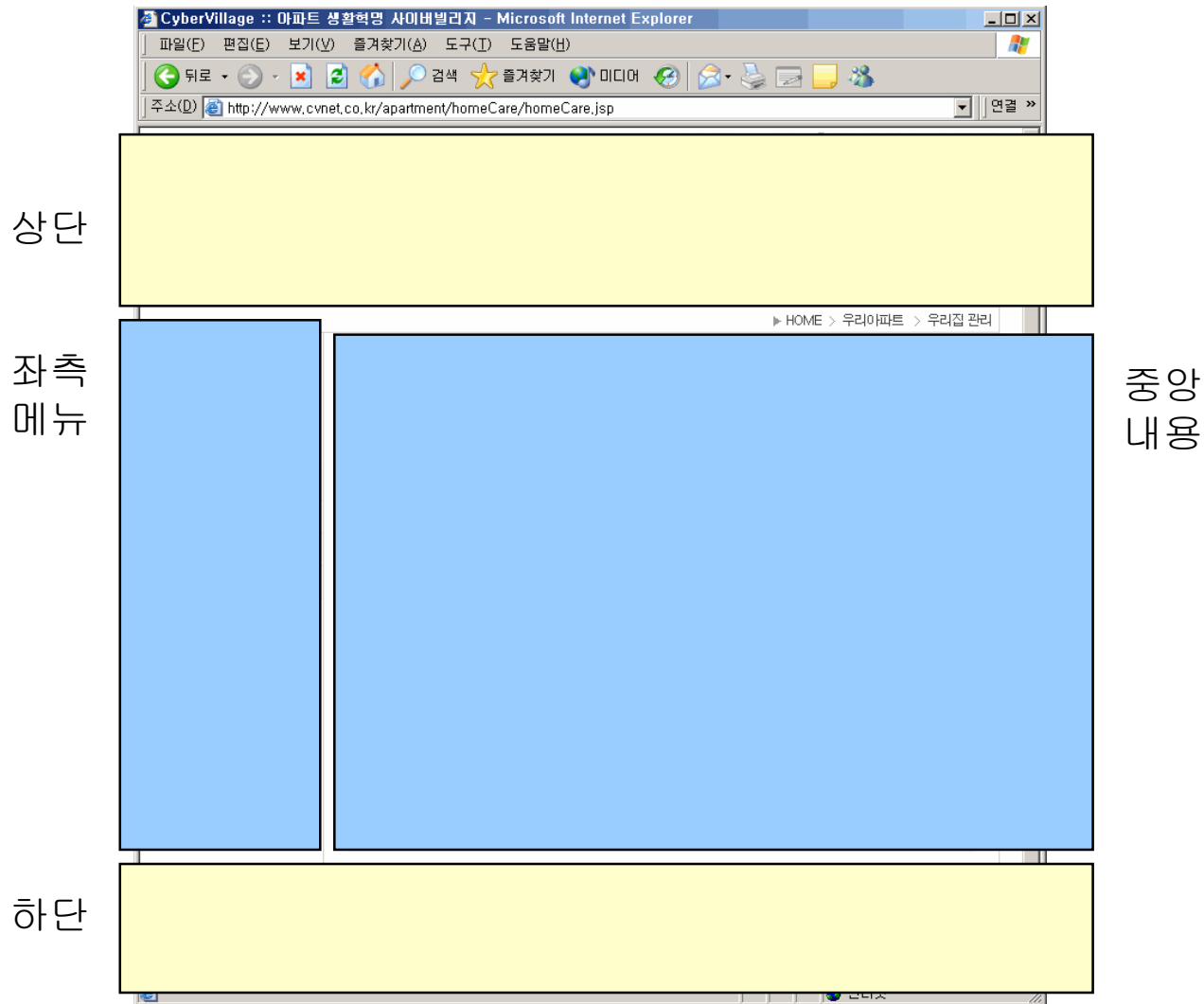
<p>
sub.jsp에서 생성한 내용.
</p>
```



sub.jsp의 결과가
<jsp:include> 위치에 포함됨

main.jsp
sub.jsp

<jsp:include>와 페이지 모듈화: 일반적 페이지 구성



공통부분

변경부분

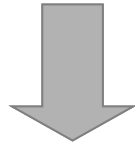
공통UI를 모듈화하지 않은 경우의 문제점!

JSP1
상단코드
...
...
...
하단코드

JSP2
상단코드
...
...
...
하단코드

JSP3
상단코드
...
...
...
하단코드

공통된 부분을 변경하려면



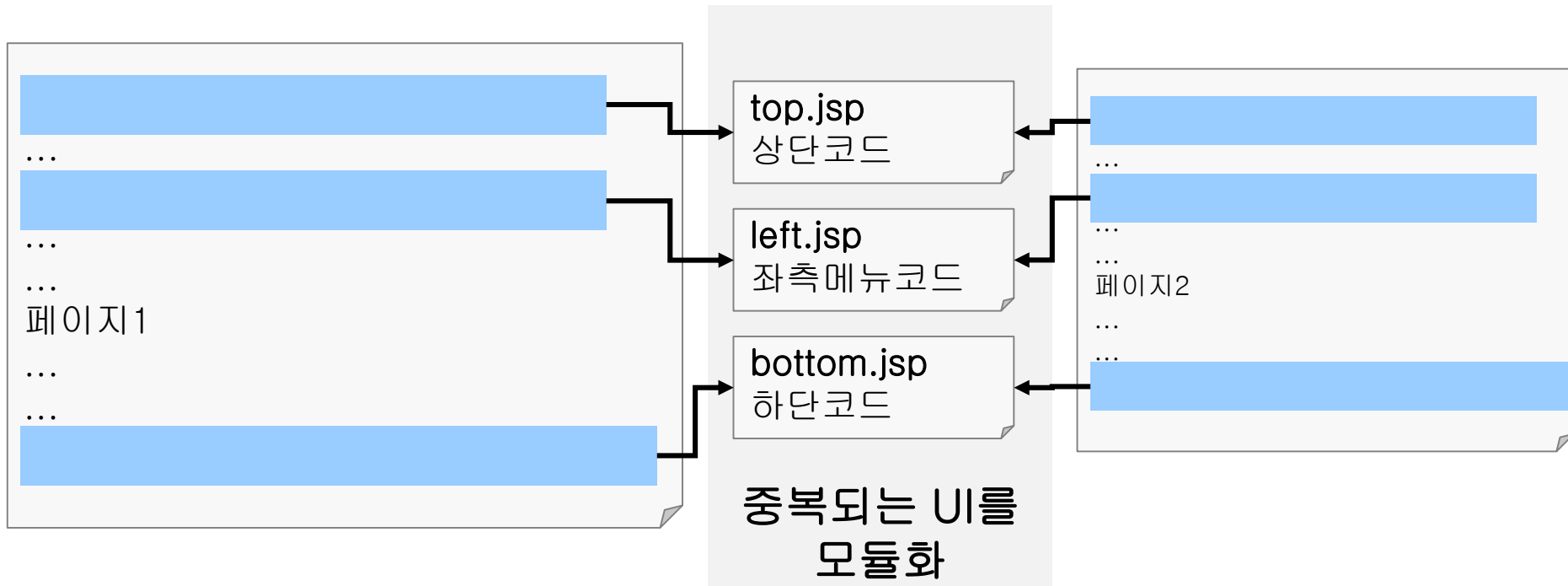
JSP1
...
...
...
하단코드

JSP2
...
...
...
하단코드

JSP3
...
...
...
하단코드

관련된 JSP 페이지를
모두 변경해주어야 한다!!

<jsp:include> 액션 태그를 이용한 페이지 모듈화

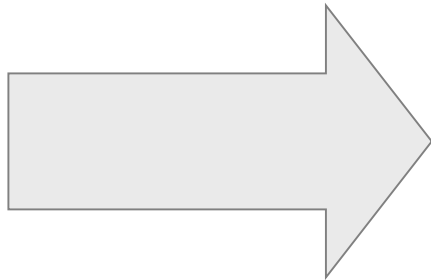


- 새로운 페이지 제작시 중복된 코드를 재작성할 필요가 없으므로 페이지 작성 시간이 단축된다. (변경되는 부분만 작성하면 된다.)
- 공통 영역을 변경해야 할 경우, 모듈화한 JSP만 변경하면 되므로 공통 영역의 유지/보수 작업이 간단해진다.

layout.jsp, layout2.jsp
module폴더파일들

공통 영역을 모듈화해도 여전히 남은 문제는?

***전체적으로 레이아웃을 변경해야 할 경우
관련된 모든 JSP 페이지를 변경해주어야 한다!!!***



JSP 레이아웃 템플릿으로
이 문제를 해결할 수 있다!

JSP 레이아웃 템플릿의 구성

template.jsp는 레이아웃과 관련된 코드를 갖고 있으며, useTemplate.jsp가 지정한 JSP를 내용 부분에 포함한다.

useTemplate.jsp
메인페이지지정

useTemplate.jsp는 템플릿 JSP를 구성하는 데 필요한 정보를 지정한다. 주로 내용 부분에 포함시킬 JSP를 지정한다.

<jsp:forward>

template.jsp

<table>

...

...

...

...

...

...

top.jsp
상단코드

left.jsp
좌측메뉴코드

bottom.jsp
하단코드

UI 모듈

실제 내용을 출력하는 JSP

내용페이지
내용코드

JSP 레이아웃 템플릿 예제

Example

```
<%@ page contentType = "text/html; charset=euc-kr" %>
```

```
<%
```

```
String pageTitle = (String)request.getAttribute("PAGETITLE");
```

```
String contentPage = request.getParameter("CONTENTPAGE");
```

```
%>
```

```
<html><head><title><%= pageTitle %></title></head><body>
```

```
<table width="400" border="1" cellpadding="2" cellspacing="0">
```

```
<tr>
```

```
<td colspan="2"><jsp:include page="/module/top.jsp" flush="false" /></td>
```

```
</tr>
```

```
<tr><td width="100" valign="top"><jsp:include page="/module/left.jsp" flush="false" /></td>
```

```
<td width="300" valign="top">
```

```
<!-- 내용 부분: 시작 -->
```

```
<jsp:include page="<%= contentPage %>" flush="false" />
```

```
<!-- 내용 부분: 끝 -->
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2"><jsp:include page="/module/bottom.jsp" flush="false" /></td>
```

```
</tr>
```

```
</body>
```

```
</html>
```

템플릿을 구성하는데
필요한 정보를 읽어온다.

내용부분은 이 템플릿 JSP를
사용하는 JSP에서 request의
속성으로 지정한다.

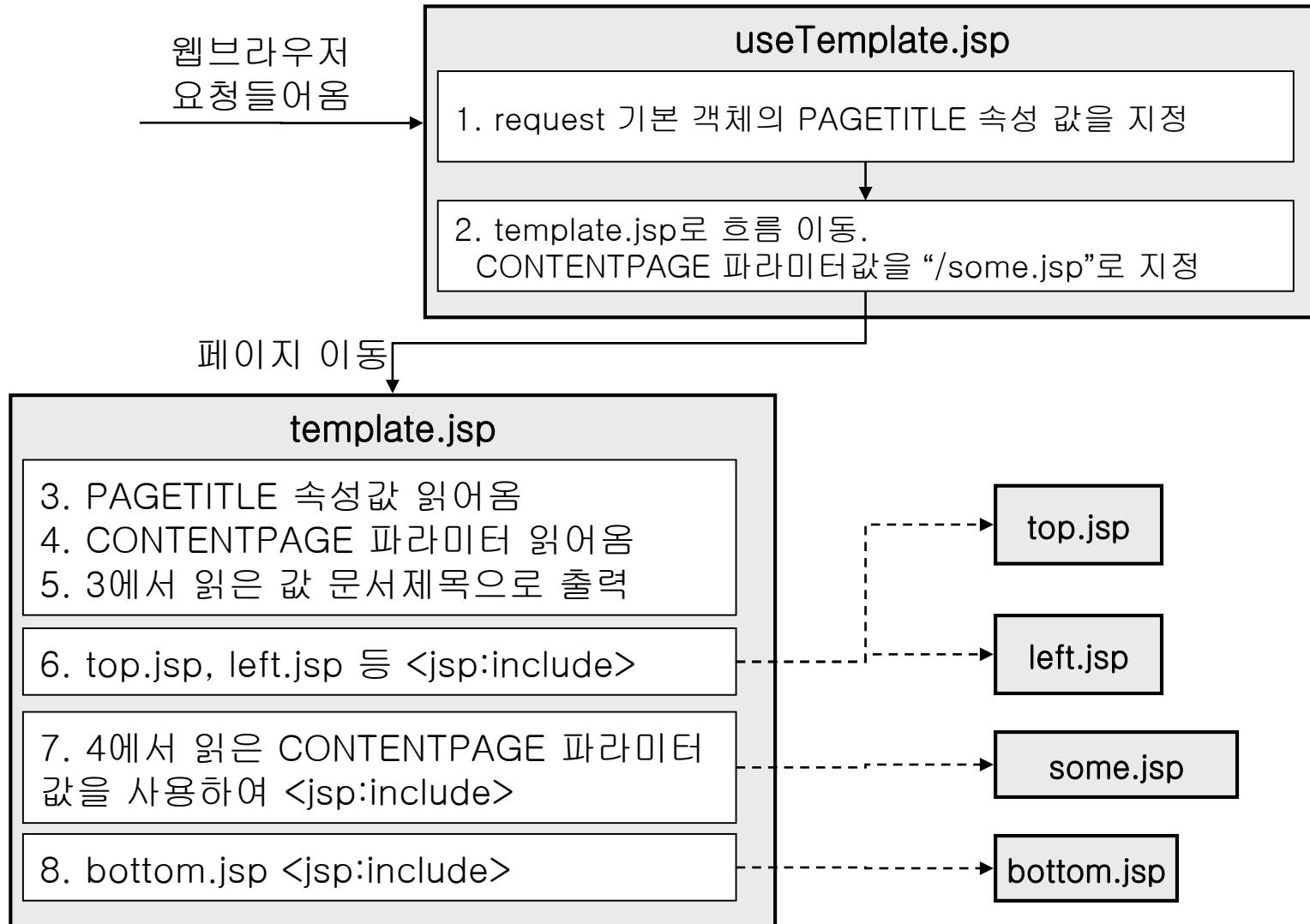
템플릿은 레이아웃과 관련된 HTML 태그를 포함한다.

템플릿을 사용하는 JSP 예제

Example

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%
    request.setAttribute("PAGETITLE", "정보보기");
%>
<jsp:forward page="/template/template.jsp">
    <jsp:param name="CONTENTPAGE" value="/info_view.jsp" />
</jsp:forward>
```

템플릿 사용시 제어 흐름



일반 방식과 템플릿 방식의 비교

일반방식	템플릿방식
<ul style="list-style-type: none">• 하나의 JSP 페이지에서 모든 것이 처리되는 형태로서 단순하다.• 실행 과정에서 문제가 발생할 때 하나의 JSP 페이지만 검사하면 된다.• 각각의 페이지가 레이아웃과 관련된 코드를 중복해서 포함하고 있다.• 레이아웃 변경시 관련된 모든 JSP 페이지를 변경해주어야 한다.	<ul style="list-style-type: none">• 레이아웃 정보를 담고 있는 템플릿 JSP 페이지가 존재한다.• 하나의 기능을 구현하기 위해서는 레이아웃 정보를 지정해주는 JSP 페이지와 내용 부분에 출력될 JSP 페이지가 필요하다.• 레이아웃 변경시 템플릿 JSP를 변경해주면 된다.

include 디렉티브

Syntax

`<%@ include file="포함할파일" %>`

* JSP의 현재 위치에 file 속성에 명시한 파일의 내용을 삽입한다.

- include 디렉티브는 지정한 파일의 내용을 현재 위치에 포함시켜주는 기능을 제공한다.
- `<jsp:include>`와 달리 include 디렉티브는 다른 파일의 내용을 현재 위치에 삽입시킨 후에 JSP 파일을 자바 파일로 변환하고 컴파일하는 방식이다.

include 디렉티브의 처리 방식

main.jsp

코드1

```
<%@ page file="sub.jspf" %>
```

코드2

sub.jspf

코드sub1

main.jsp의 page 디렉티브 위치에
포함할 페이지의 코드를 삽입시킨 결과를
자바코드로 작성한다.

자바코드

코드1

코드sub1

코드2

자바코드를 서블릿 클래스로 컴파일한다.

서블릿 클래스

...

include 디렉티브
부분에 file 속성에
명시한 파일이 삽입

include 디렉티브 예제 - 처리 전

Example

includer.jsp

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

<%
    int number = 10;
%>

<%@ include file="/includee.jsp" %>

공통변수 DATAFOLDER = "<%= dataFolder %>"

</body>
</html>
```

includee.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
includer.jsp에서 지정한 번호: <%= number %>
<p>
<%
    String dataFolder = "c:WWdata";
%>
```


include 디렉티브 예제 - 처리 결과

Example

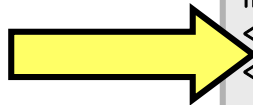
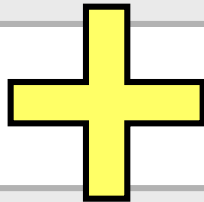
```
<%@ page contentType = "text/html; charset=euc-kr" %>
<html>
<head><title>include 디렉티브</title></head>
<body>

<%
    int number = 10;
%>

<%@ include file="/includee.jsp" %>

공통변수 DATAFOLDER = "<%= dataFolder %>"

</body>
</html>
```



```
<%@ page contentType = "text/html; charset=euc-kr" %>
includer.jsp에서 지정한 번호: <%= number %>
<p>
<%
    String dataFolder = "c:WWdata";
%>
```

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

<%
    int number = 10;
%>

includer.jsp에서 지정한 번호: <%= number %>
<p>
<%
    String dataFolder = "c:WWdata";
%>

공통변수 DATAFOLDER = "<%= dataFolder %>"

</body>
</html>
```

include 디렉티브의 활용

- 모든 JSP 페이지에서 사용되는 공용 변수 지정
- 저작권 표시와 같은 간단하면서도 모든 페이지에서 중복되는 문장

Example

commonVariable.jspf

```
<%  
    String memberId = (String)session.getAttribute("MEMBERID");  
    File tempFolder = (File)application.getAttribute("TEMPFOLDER");  
%>
```

공통변수를 사용하는 JSP

```
<%@ include file="commonVariable.jspf" %>  
...  
<%= memberId %>  
...
```

<jsp:include> 액션태그와 include 디렉티브

비교항목	<jsp:include>	include 디렉티브
처리시간	요청 시간에 처리	JSP 파일을 자바 소스로 변환할 때 처리
기능	별도의 파일로 요청 처리 흐름을 이동	현재 파일에 삽입시킴
데이터 전달방법	request 기본 객체나 <jsp:param>을 이용한 파라미터 전달	페이지 내의 변수를 선언하 후, 변수에 값 저장
용도	화면의 레이아웃의 일부분을 모듈화할 때 주로 사용된다.	다수의 JSP 페이지에서 공통으로 사용되는 변수를 지정하는 코드나 저작권과 같이 공통의 문장을 포함한다.