

03. 내장객체(Implicit Object)

1) 개요

: JSP 프로그램 코드에서 프로그래머가 선언하지 않아도 사용할 수 있는

미리 선언된 객체

: `<% %>` ScriptLet 안에서만 사용이 가능하다. → `<% %>` 열고, 닫고 중요

ex) response / request

2) 내장객체의 생성

: Servlet Container 가 해당 JSP 페이지 실행 시 **자동으로** 생성해준다.

3) 내장객체가 선언되는 곳


: JSP 실행 시 자동 생성되는 **Servlet 코드**의 `_jspService() method()` 내에 선언된다.

4-1) request 객체 : request 객체의 활용

```
<body>
<%!
    String name ="홍길동";
%>
<!-- request객체 : 사용자로부터 요청을 받았을때 사용되는 jsp의 내장객체 -->
<h2>
1. 서버의 도메인이름: <%=request.getServerName() %> <br>
2. 서버의 포트번호: <%=request.getServerPort() %> <br>
**3. 요청된 URL에서의 경로: <%=request.getRequestURL() %> <br>
**4. 요청된 URI에서의 경로: <%=request.getRequestURI() %> <br>
**5. 원격 주소 ip 주소: <%=request.getRemoteAddr() %> <br>
6. 클라이언트 요청 프로토콜: <%=request.getProtocol() %> <br>
**7. 요청방식(Get or Post): <%=request.getMethod() %> <br>

</h2>

</body>
```



1. 서버의 도메인이름: localhost
2. 서버의 포트번호: 8090
**3. 요청된 URL에서의 경로:
http://localhost:8090/JspWork/inobject/req.jsp
**4. 요청된 URI에서의 경로: /JspWork/inobject/req.jsp
**5. 원격 주소 ip 주소: 0:0:0:0:0:0:1
6. 클라이언트 요청 프로토콜: HTTP/1.1
**7. 요청방식(Get or Post): GET

4-2) response 객체 : response 객체의 활용

```
<title>응답객체(=response)</title>
```

```
</head>
```

```
<body>
```

```
<%
```

```
response.setContentType("text/html; charset=utf-8");
```

문서의 타입설정 가장 상단에 선언된 문장이

➔ 서블릿으로 바뀌면서 설정된다.

```
response.sendRedirect("http://www.daum.net");
```

➔ 멈추지 않고 바로 외부의 사이트로 이동함(URL내용이 바뀜)

```
response.sendRedirect("../bottom/bottom.html");
```

➔ 내부의 특정 위치로 이동

형식) response.sendRedirect("경로포함 이동할 페이지명");

```
%>
```

```
<script> // Javascript 로 대체할 경우
```

```
//location.href="http://www.naver.com";
```

```
location.replace("http://www.nate.com"); // replace함수 : url경로 변경
```

```
</script>
```

※ cf) Web 프로그래밍에서 페이지를 이동시키는 방법 ver. JSP (면접시 질문)

1.response객체 사용

response.sendRedirect("http://www.chosun.com"); 외부의 사이트로 이동
response.sendRedirect("./req.jsp"); //상대경로를 지정(내부 파일이동)

- **URL창이 이동할 페이지로 전환이 되면서 이동이 된다.
- 데이터를 서로 공유할 수 없다.(데이터를 공유하면서 이동X)

2.forward 액션태그 → 모델2의 핵심

- 외부의 사이트로 이동X
- 자기 프로젝트의 다른 페이지로만 이동이 가능하다.(O)

→ **URL창이 이동할 페이지로 전환이 X **

∴ 데이터를 공유하면서 페이지를 이동시킬 수 가 있다.

회원로그인->예매페이지 1-->예매페이지 2--->예매페이지 3
지역 상영관 시간 카드,무통장입금

cf) 웹->페이지이동 하면 전의 정보가 사라진다 ?!

→ 세션처리를 통해서 유지할 수 있다.

3.자바스크립트->location.href="이동할 페이지명"

location.replace("http://~이동할 페이지명")

history.back()->전의 페이지로 이동

history.go(-1)->전의 페이지로 이동

history.forward()->앞의 페이지로 이동

04. 액션태그 (Action Tag)

1) 개요

: 어떤 기능(페이지 이동, 외부의 파일을 불러오는)을 가진 태그

ex) index.jsp 에서

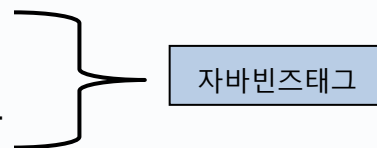
→ top.jsp / bottom.jsp / left.jsp / body.jsp 를 보여줄수 있는 기능

: XML 태그와 같은 모양을 가지며, JSP 에서 제공하는 특별한 기능을 사용하기 위한 태그

실행 시 자바 코드로 자동 변환된다. → JSP 코드를 보다 쉽고 빠르게 작성하기 위해 사용

2) 종류

- ① **useBean** : JSP 페이지 내에서 자바빈즈 객체 생성
- ② **setProperty** : 자바빈즈 객체 멤버변수 값 지정
- ③ **getProperty** : 자바빈즈 객체의 특정 멤버변수 값을 반환
- ④ **include** : 다른 JSP 페이지를 호출하여 실행 한 후 돌아옴
- ⑤ **forward** : 다른 JSP 페이지로 처리 제어권을 넘김



3) include 액션태그 / include 지시어

(1) 공통점

: 둘 다 외부의 파일을 불러와서 특정한 위치에 삽입한다.

(2) 차이점

: include 액션태그 → 동적으로 변경된 내용을 특정한 위치에 삽입 (실행결과를 봐야)

→ 컴파일이 되서 삽입된다!

형식) `<jsp:include page="이동페이지" flush="false">`

: include 지시어 → 정적으로 문자열만 복사하여 특정한 위치에 삽입

→ 코드가 삽입된다고 생각하고, 컴파일은 한 파일에서 진행된다.

형식) `<%@include file ="경로포함하여 파일명" %>`

form.jsp (include 액션태그)

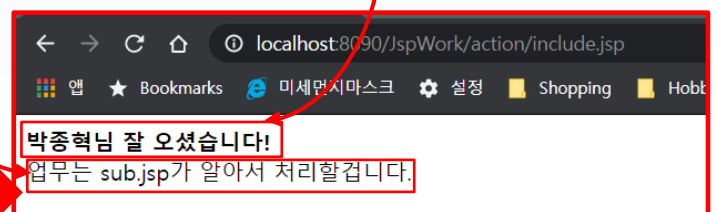
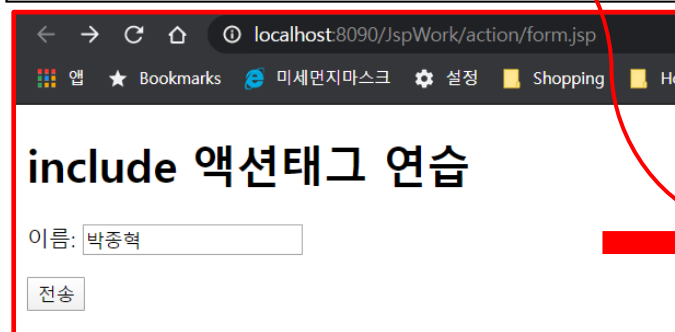
```
<body>
<h1>include 액션태그 연습</h1>
<form action="include.jsp" method="post">
<p>이름: <input type="text" name="name"></p>
<p><input type="submit" value="전송"></p>
</form>
</body>
```



include.jsp (include 액션태그)

sub.jsp

<pre><body> <% request.setCharacterEncoding("utf-8"); %> <!-- sub.jsp에게 요청해서 실행한 결과 내용을 삽입--> <jsp:include page="sub.jsp" flush="false"/>
 <!-- sub.jsp --> 업무는 sub.jsp가 알아서 처리할 겁니다. </body></pre>	<pre><body> <!-- sub.jsp --> String name = request.getParameter("name"); %> <%=name%> 님 잘 오셨습니다! </body></pre>
--	--



<식별자:액션태그명 액션태그의 속성명="속성값(경로포함해서 이동할페이지명)"~ />

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

- 속성명 순서는 정해져 있지 않음(순서 중요X)
- 식별자 → 액션태그의 종류를 구분하는 인자로 사용된다.

includeTest.jsp (include 지시어)

JspModule.java (servlet/ include 지시어)

<pre><body> <%@include file="./sub/JspModule.java" %> <h3>이 페이지는 include지시어를 연습하기위해서 만들어진 페이지입니다.</h3> <h3>테스트중입니다.</h3> <%=getStr() %> <h3>테스트중입니다.</h3> <%=getStr() %> <h3>테스트중입니다.</h3></pre>	<pre><%! // 클래스 이름이 정해져 있지 않음 → 정적메소드 처럼 불러올 수 있다. String str = "Testing practice"; String getStr(){ return str; } %></pre>
--	---

컴파일이 되지 않은 상태이기 때문에 빨간줄이 가면서 에러표시가 난다
실행시키면, 컴파일 되면서 getStr()을 사용할 수 있기 때문에 문제발생이 되지 않는다.

4) forward 액션태그 : <jsp:forward page= "이동할 페이지">

JSP 페이지 내에 forward 액션태그를 만나면 그 전까지 출력 버퍼에 저장되어 있던 내용을 제거하고, forward 액션태그가 지정하는 페이지로 이동하게 된다. 사용자가 입력한 값에 따라 여러 페이지로 이동해야 할 경우 사용하면 좋다.

```
<body>
<h1>페이지를 이동시키는 역할</h1>
<form action="move2.jsp" method="post"> → getAttribute
<form action="move.jsp" method="post"> → getParameter
  <p>이동할 페이지명: <input type="text" name="move"> </p>
  <p><input type="submit" value="전송"> </p>
</form>
</body>
```

move.jsp 로 넘길경우

```
<body>
<%
    String move = request.getParameter("move"); //a.jsp or b.jsp%>
<!-- forward 액션태그 page = "이동할페이지명"-->
<jsp:forward page = "<%=move%>" /> <br>
</body>
```

//서버에 저장된 값을 불러오는 것이 아님 -> Get방식
request.getParameter("전달받은 매개변수");

<body>
<%

move2.jsp
로 넘길
경우

```
int su = 100;
System.out.println("su:"+su);

Date d = new Date(); //-> b.jsp로 전달하여 출력
// 자바의 자료구조 -> HashMap(키, 값)
// 형식) request|session|application.setAttribute(key, 값)
--> 서버로 넘겨서 실제로 저장 되는 구문 : setAttribute
request.setAttribute("total",su); //-> new Integer(su) 지만 자동 형변환 Wrapper
request.setAttribute("cal",d); //request.getAttribute("cal");
//-----
String move = request.getParameter("move"); //a.jsp or b.jsp
//이동시킬 페이지가 존재하면 -> 이동 0 or 존재 x -> 다시 입력을 받을 수 있도록 코딩
if(move.equals("a.jsp")){
    //reponse.sendRedirect("a.jsp"); ==> 이동만 하는 역할이라면 이대로도 됨
    <!-- forward 액션태그 page = "이동할페이지명"-->
    <jsp:forward page = 'a.jsp' /><br>
<%}else (if move.equals("b.jsp")){%>
<jsp:forward page = 'b.jsp' /><br>
<%}else{ %>
<script>
    alert("페이지가 존재하지 않습니다. 다시 입력해주세요");
    location.href="forward.jsp";
</script>
<%} %>
</body>
```

//서버에 저장된 값을 불러오는 것
request.getAttribute ("서버에 저장된 키명");

a.jsp

```
<body bgcolor="#09f">
<h1>a.jsp에 오신것을 환영합니다.</h1>
<%
int su = 200;
// 자기것 -> page 범위 (일종의 지역변수개념)
%>
<hr>
move2.jsp에서 공유한 su의 값을 찍어라
<%=request.getAttribute("total") %>

</body>
```

b.jsp

```
<body bgcolor="#bf9">
<h1>b.jsp에 오신것을 환영합니다.</h1>
<hr>
move2.jsp에서 공유한 날짜객체(d)의 값을 찍어라
<%=request.getAttribute("cal") %>
</body>
```

//서버에 저장된 값을 불러오는 것
request.getAttribute ("서버에 저장된 키명");

****a.jsp (int su=100;) → 현재 보고 있는 페이지에서만 변수사용(default page)**

b.jsp-> request(요청 → 응답) → 하나의 변수를 두 페이지에서 공유해서 사용

5) 웹상에서 내장객체 변수의 사용범위

(1) **page** 범위 → 한개의 jsp 에서 사용하는 범위

(자기 자신의 페이지에서만 사용이 되는 변수범위) → 공유 X

(2) **request** 범위 → 한 페이지 이상에서 공유해서 사용하는 경우

request.setAttribute(키, 값) → move.jsp

request.getAttribute(키명) → a.jsp or b.jsp

(3) **session** 범위 => 최소 한페이지 이상에서 공유

회원로그인 한 동안 공유하는 페이지를 작성 → 이 개념으로 생각
(로그아웃할때 까지 공유->ex 7 페이지)

session.setAttribute(키, 값)

session.getAttribute(키명)

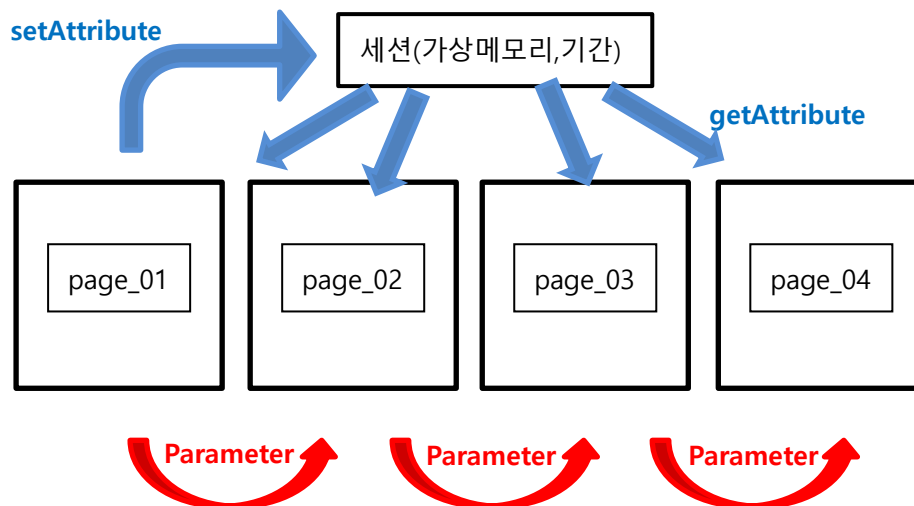
(4) **application** 범위 => 모든 페이지에서 공유해서 사용할 경우

ex) 30 페이지->전부 공유

application.setAttribute(키, 값)

application.getAttribute(키명)

cf) set/getAttribute 와 set/getParameter 의 개념



(1) Attribute 방식

: 웹의 메모리상에 일정 시간동안 사용하려는 값을 올려두고, 각 페이지에서 공유하여 쓸수 있도록 하는 방식

(2) Parameter 방식

: 매번 다른 페이지로 전달을 해야만 하는 방식

※ 뒤에 나오는 자바빈즈 객체의 set/getProperty 는 객체로 DB 에 접근하여 값을 가져오는 방식으로 위 개념과 관계가 없음. 헛갈리지 말 것!

※ 메인페이지를 작성하는 방법 (forward 태그 이용)

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>페이지 전달용</title>
</head>
<body>
<!-- index.jsp ->
<jsp:forward page="/view/indexControl.jsp">
    <jsp:param name="CONTROL" value="intro"/>
    <jsp:param name="PAGENUM" value="01"/>
</jsp:forward>
<!-- 액션 태그에는 눈에 보이지 않는 주석을 주어야 정보는닉이 가능하다. %-- --% 로 -->
<%--<jsp:forawrd or include page ="상대경로">
<jsp:param name="CONTROL" value="intro"/>
<jsp:param(액션태그명) name="매개변수명" value="전달할 값">--%>
</body>
</html>
```

※ 매개변수를 전달하는 방식

① 형식

```
<jsp: forward page=" ">
    <jsp:param name=" " value=" " /> → parameter로 넘길 인자
</jsp:forward>
```

② /view/indexControl.jsp → jsp파일의 경로

/view/indexCotrol.jsp?CONTROL (매개변수명)=값&(매개변수2)=값2&

indexControl페이지로 넘기는데, 파라미터는 CONTROL, PAGENUM 두개를 넘김.
그 값은 intro 01 페이지

indexControl.jsp : forward 를 통해서 indexControl 로 parameter 가 전달

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!-- 디자인 목적이 아니어서 html 필요없음 --><%
    // /view/indexControl.jsp?CONTROL=intro&PAGENUM=01 ==> URL창에 노출되는 내용
    String CONTROL = null;
    String PAGENUM = null;
    try{
        CONTROL = request.getParameter("CONTROL");
        PAGENUM = request.getParameter("PAGENUM");
        // setAttribute() , getAttribute 와 getParameter는 다르다
        // 구분하여 사용할것, Attribute 는 set으로 서버에 저장하고 get으로 불러오는 것

        //intro_01.jsp를 못받을 수도 있음
        if(CONTROL.equals(null)){
            CONTROL = "intro";
        }
        if(PAGENUM.equals(null)){
            PAGENUM = "01";
        }
    }catch (Exception e){
        e.printStackTrace();
    }
%>
<jsp:forward page="/view/template.jsp">
    <jsp:param name="CONTROL" value ="<%=CONTROL %>" />
    <jsp:param name="PAGENUM" value ="<%=PAGENUM %>" />
</jsp:forward>
```

template.jsp 로 parameter를 넘기면서 페이지를 이동

최종적으로 index.jsp → indexControl.jsp ->
template(intro_01.jsp) 를 여는 것과 같다

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%

    String CONTROL = request.getParameter("CONTROL");
    String PAGENUM = request.getParameter("PAGENUM");

    // Left메뉴의 jsp 파일명조합을 보면 Left로 조합이 되어있다.
    String left="/module/"+CONTROL+"Left.jsp"; // 경로를 left 라는 변수에 저장
    // 본문내용 -> /view/intro_01.jsp
    String content="/view/"+CONTROL+"_"+PAGENUM+".jsp" // 파일명

%>
<html>
<head>
<meta charset="UTF-8">
<title>실질적인 메인페이지</title>
</head>
<body>
    <table width="800px" height="600px" border="1" cellpadding="2" cellspacing="0">
        <!-- 상단메뉴 -->
        <tr height="50">
            <td colspan="2">
                <jsp:include page="/module/top.jsp" flush="false"/> // ①
            </td>
        </tr>
        <!-- 좌측메뉴 , 본문 -->
        <tr height="500px">
            <td width="150" valign="top">
                <jsp:include page="<%=left %>" flush="false"/> // ②
            </td>
            <td width="650" valign="top">
                <jsp:include page="<%=content %>" flush="false"/> // ③
            </td>
        </tr>
        <!-- 하단메뉴 -->
        <tr height="50px">
            <td colspan="2">
                <jsp:include page="/module/bottom.jsp" flush="false"/> //④
            </td>
        </tr>
    </table>

```

jsp:include page로
각 영역에 뿌려줌

template.jsp 출력형태

localhost:8090/JspWork2/view/indexControl.jsp?CONTROL=intro&PAGENUM=01

회사소개 제품소개 연구개발 묻고답하기 자료실

① - `/module/top.jsp`

회사소개
인사말
연혁
비전
찾아오시는길

② - `<%=left%>`

저희 회사를 홈페이지를 방문해 주셔서 감사합니다.

③ - `<%=content%>`

실제로 변하는 페이지는 2,3번

개인정보 보호 정책 | 홈페이지 이용 안내 | E-MAIL | 도움말 | 약관 |

본 웹사이트는 이메일 주소의 무단 수집을 금지합니다.

④ - `/module/bottom.jsp`