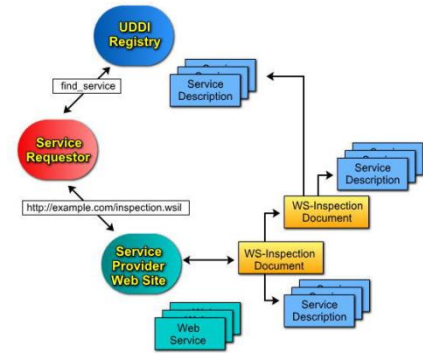


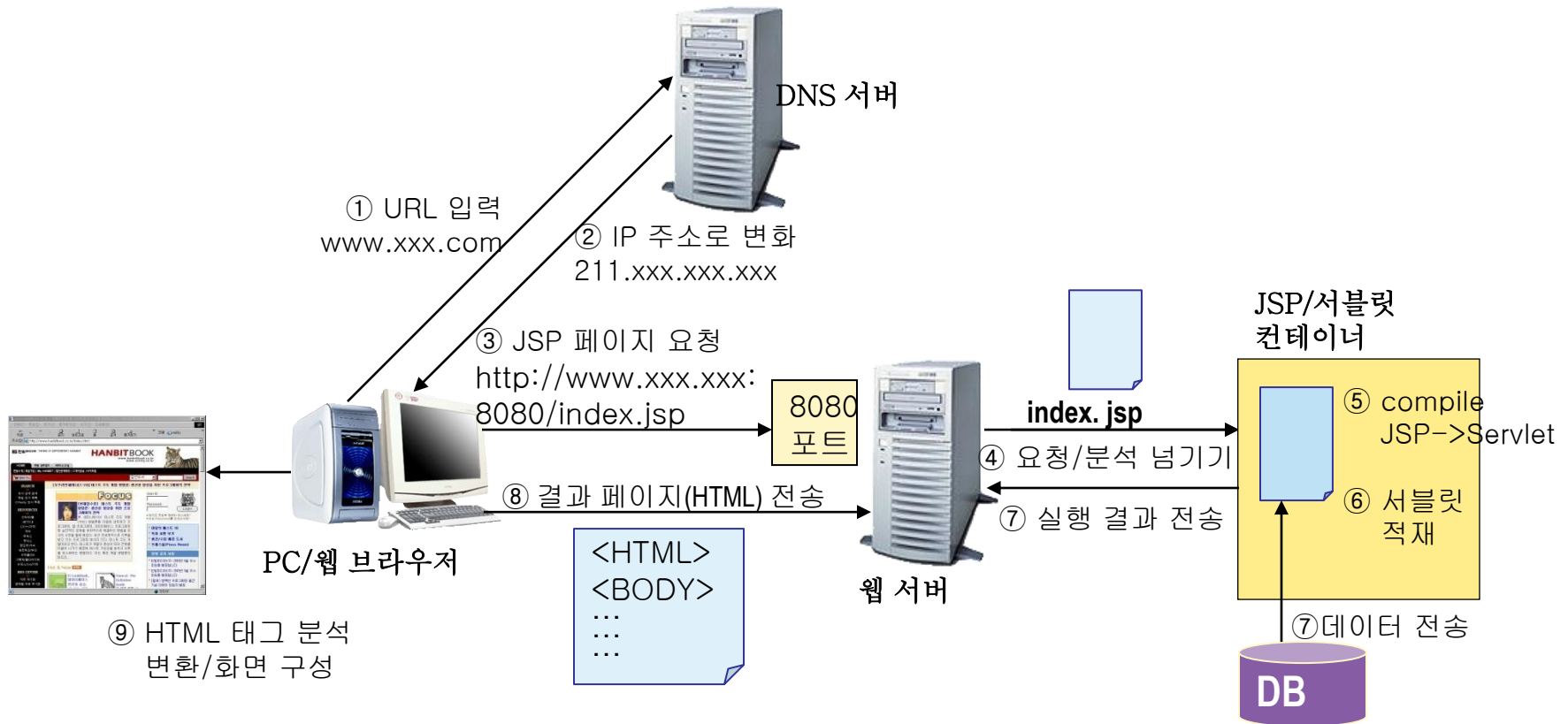
Web Application & Service

JSP 문법

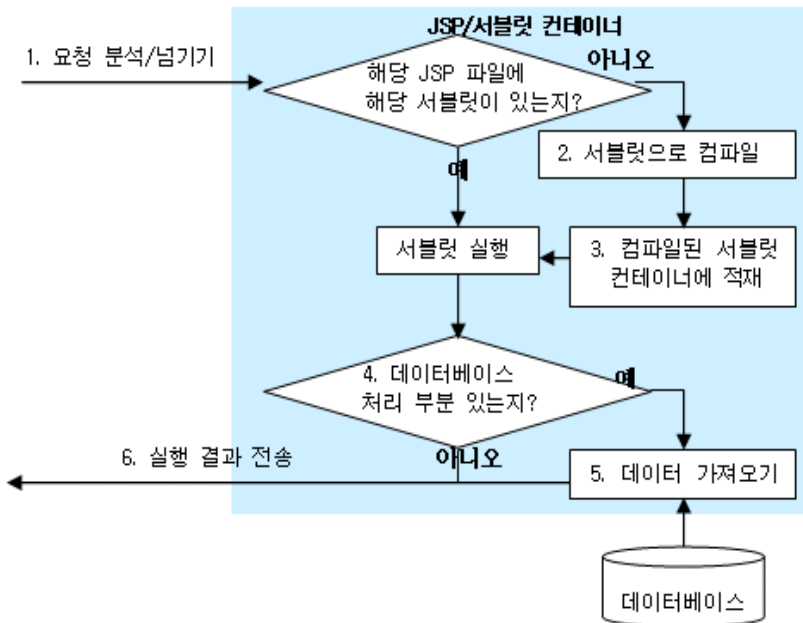
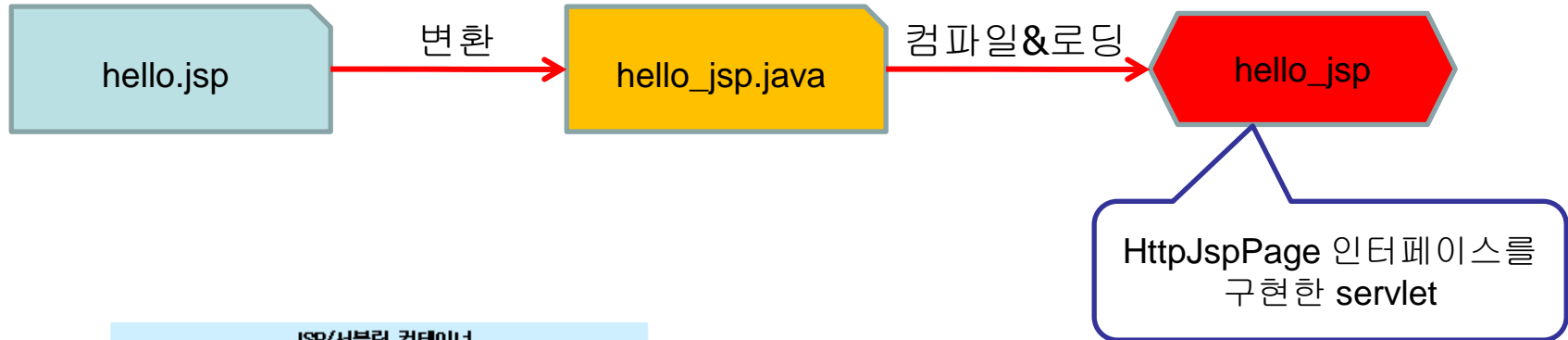


컴퓨터 과학과
김희열

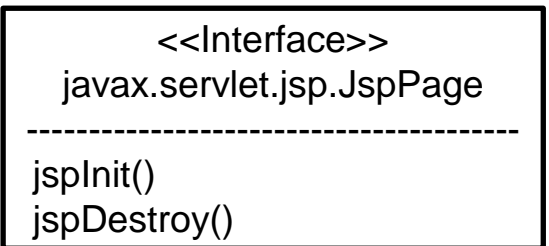
JSP



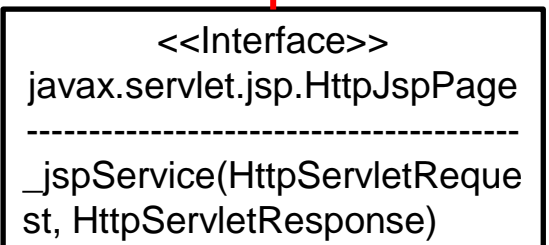
Servlet으로의 변환



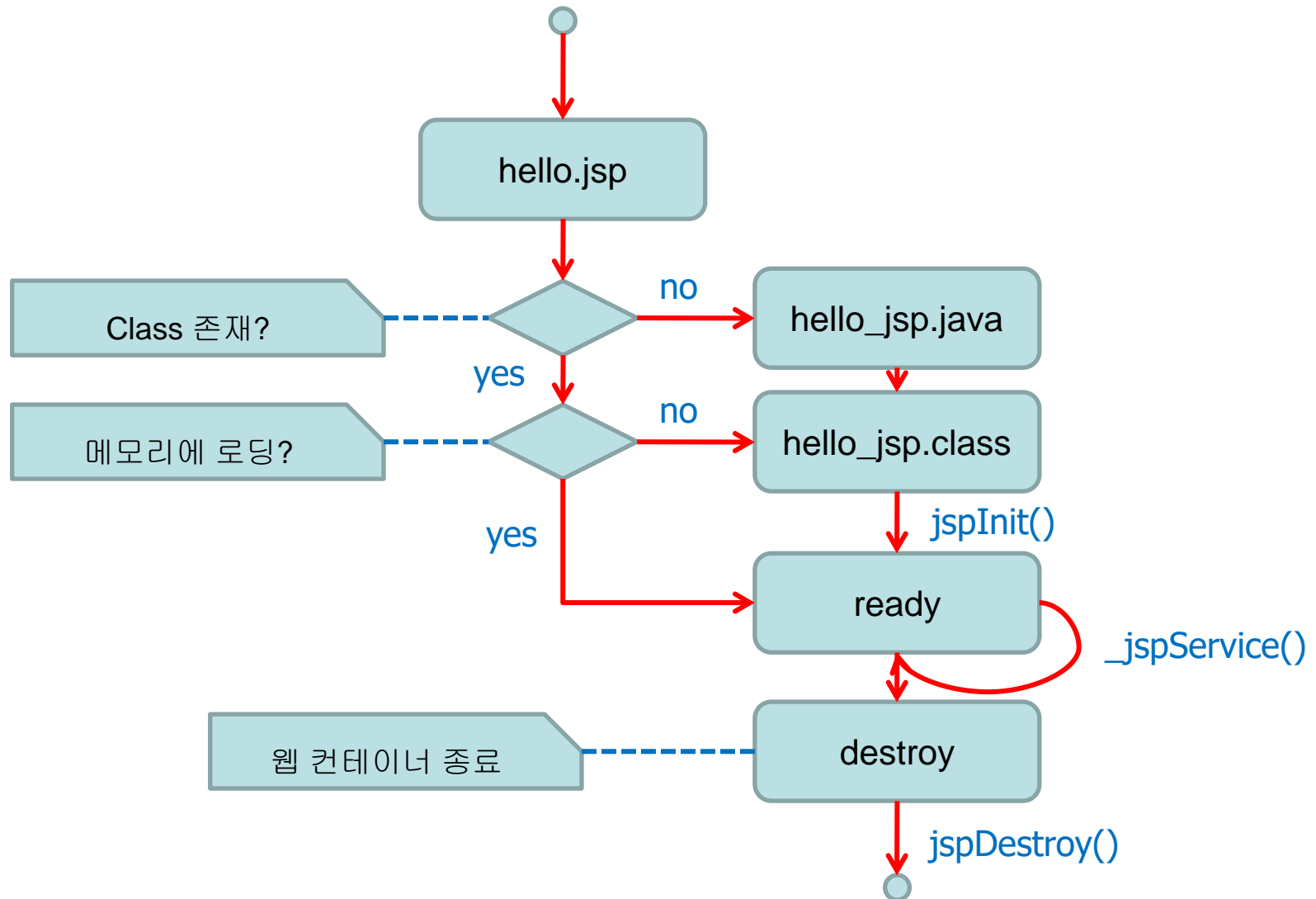
재정의 가능



재정의 불가능



Servlet으로의 변환



Servlet으로의 변환

- 변환 예
 - HttpJspPage를 구현한 실제 servlet 클래스는 container마다 다름

```
public final class Hello_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

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

public abstract class **HttpJspBase** extends [javax.servlet.http.HttpServlet](#)
implements [javax.servlet.jsp.HttpJspPage](#)

주석

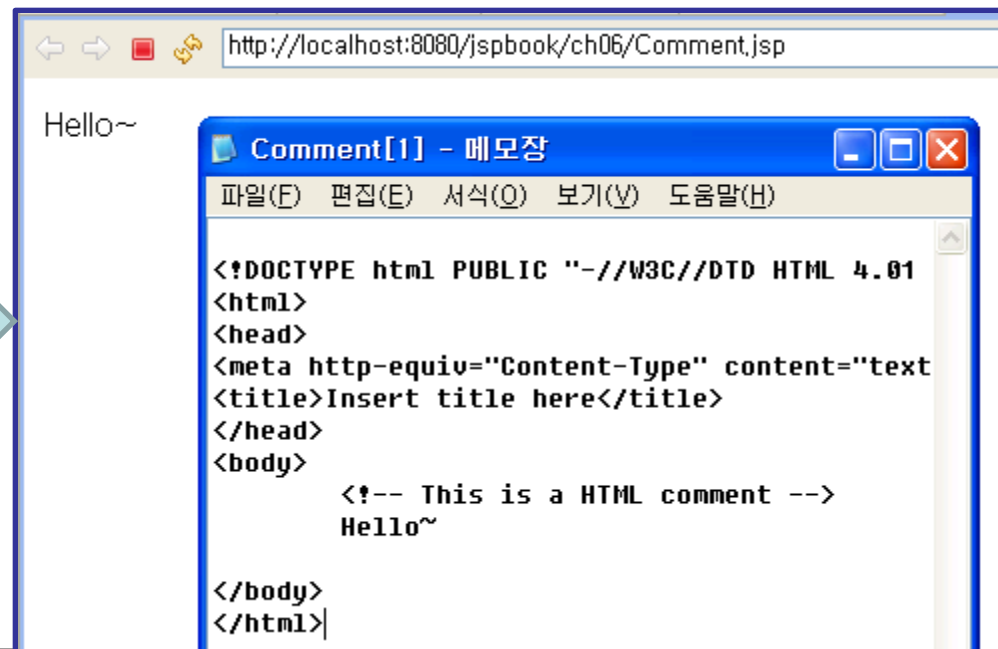
- HTML 주석

- <!-- HTML 주석 -->
- 클라이언트에게 전송되어 브라우저에 의해 처리

- JSP 주석

- <%-- JSP 주석 --%>
- 서버 내부에서만 처리되는 주석

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html">
<title>Insert title here</title>
</head>
<body>
    <!-- This is a HTML comment -->
    Hello~
    <%-- This is a JSP comment --%>
</body>
</html>
```



지시어 (directives)

- JSP 파일의 속성을 기술
- Container에게 해당 파일을 어떻게 처리해야 하는지 전달
- Directives
 - page
 - include
 - taglib

Page Directives

- 현재 JSP 페이지의 속성을 기술
- `<%@ page 속성1="속성값1" 속성2="속성값2" ... %>`

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"  
    pageEncoding="EUC-KR"%>
```

JSP 파일 생성시 자동 추가

- 속성
 - contentType
 - pageEncoding
 - import
 - errorPage
 - isErrorPage
 - ...

contentType / pageEncoding

- contentType

- JSP를 통해 생성된 결과물의 타입 설정
- 한글 처리를 위한 인코딩 방식도 함께 설정

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

 (in servlet)

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

 (in JSP)

- pageEncoding

- 페이지의 character set 설정
- JSP 2.0 이후 추가된 속성
- 한글 지원을 위해서는 pageEncoding과 contentType을 병행해서 사용

```
<%@ page pageEncoding="UTF-8" %>
```

import

- Java 언어의 **import**와 동일한 기능
- JSP 내에서 외부 패키지와 클래스를 사용해야 하는 경우 기술
- 기본적으로 포함되는 패키지
 - java.lang.*
 - javax.servlet.*
 - javax.servlet.jsp.*
 - javax.servlet.http.*

```
<%@ page import="java.sql.*, java.util.*" %>  
<%@ page import="myapp.*" %>
```

errorPage / isErrorPage

- JSP 내에서 오류가 발생하는 경우를 처리
- errorPage
 - 오류 발생시 호출할 페이지 지정
- isErrorPage
 - 작성되는 JSP가 오류 처리용 페이지임을 명시
- 장점
 - 환경(개발/배포)에 따라 다양한 방식의 오류 처리 가능

errorPage / isErrorPage

- 기본적인 오류 처리 방식



Error_test.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Error test</title>
</head>
<body>
    <H2>Error test page</H2>
    <%= 5/0 %>
</body>
</html>
```

errorPage / isErrorPage

- errorPage / isErrorPage 활용

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR" errorPage="Error.jsp" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Error test</title>
</head>
<body>
    <H2>Error test page</H2>
    <%= 5/0 %>
</body>
</html>
```

Error_test.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR" isErrorPage="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Error</title>
</head>
<body>
    <H2>처리중 아래와 같은 에러가 발생했습니다. </H2>
    <%= exception.toString() %>
    <hr>
    contact to admin@myserver.com
</body>
</html>
```

Error.jsp

Include Directives

- 현재의 JSP 파일에 다른 HTML이나 JSP를 포함하는 기능 제공
- 장점
 - 각 기능별로 모듈화된 JSP를 작성
 - 여러 모듈을 `include`를 통해 불러들여 새로운 페이지 구성이 용이
 - 구조 변경 등의 유지보수에 용이 (프로그램 모듈화와 유사)
- 여러 페이지에 공통으로 포함되는 내용을 관리할 때 유용
 - CSS 및 JSP 공통 선언 포함
 - 저작 정보, 연락처 등 포함

Include Directives

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Include test page</title>
</head>
<body>
    <H2>This is my first page </H2>
    <HR>
    <%@ include file="Company.jsp" %>
</body>
</html>
```

Include_test.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    Last modified at 2009/10/05 by administrator of MyService.com corp. <BR>
    Email : admin@myservice.com
</body>
</html>
```

Company.jsp

http://localhost:8080/jspbook/ch06/Include_test.jsp

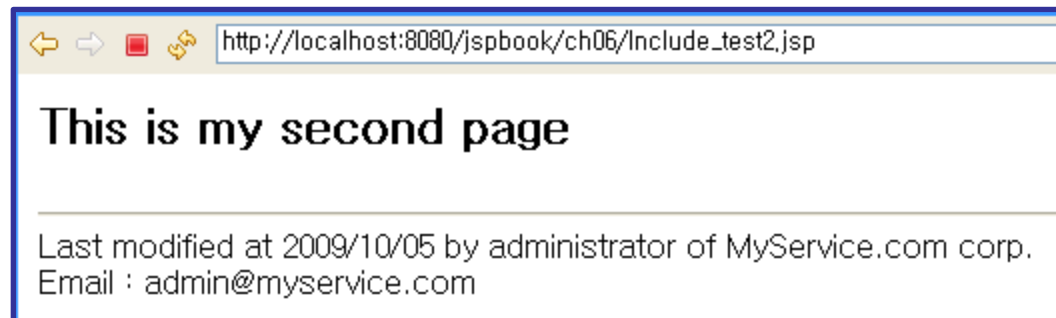
This is my first page

Last modified at 2009/10/05 by administrator of MyService.com corp.
Email : admin@myservice.com

Include Directives

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Include test page</title>
</head>
<body>
    <H2>This is my second page </H2>
    <HR>
    <%@ include file="Company.jsp" %>
</body>
</html>
```

Include_test2.jsp



Taglib Directives

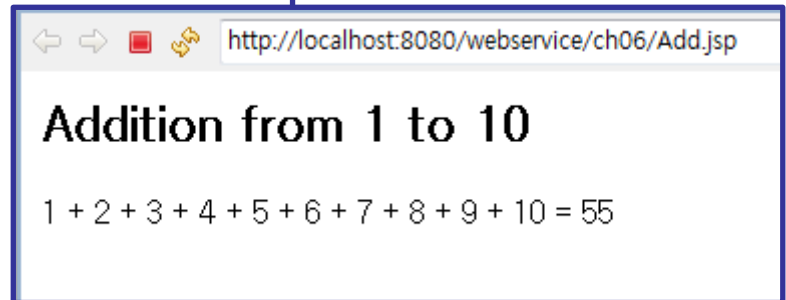
- 사용자 정의 태그 라이브러리를 사용하기 위한 정보 기술

```
<%@ taglib uri="/META-INF/mytag.tld" prefix="mytag" %>
```

Scriptlet

- JSP 내에 java코드를 적는 부분
- `<% source code ... %>`

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Scriptlet test</title>
</head>
<body>
    <H2> Addition from 1 to 10 </H2>
    <%
        int sum = 0;
        int i;
        for (i=1; i<=10; i++) {
            out.print(i + " ");
            if (i != 10) out.print("+ ");
            else out.print("= ");
            sum = sum + i;
        }
        out.println(sum);
    %>
</body>
</html>
```



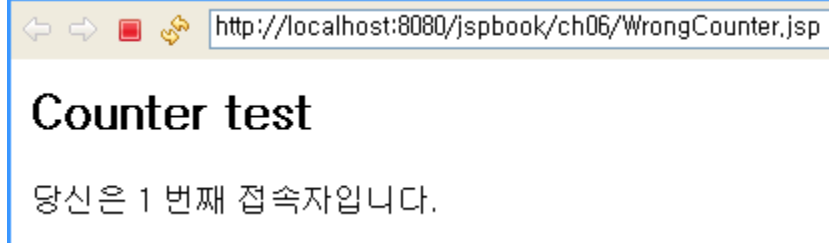
Counter JSP 만들기

- 이 페이지의 방문 횟수를 기록해서 보여주는 JSP 작성

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <H2>Counter test</H2>
    <% int count=0; %>
    <P>
    당신은 <% out.println(++count); %>번째 접속자입니다.
    </P>
</body>
</html>
```

WrongCounter.jsp

제대로 동작할까?



- 생성된 servlet

_jspService()내의
로컬 변수로 선언

```
public final class WrongCounter_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

        ...

        try {
            _jspxFactory = JspFactory.getDefaultFactory();
            response.setContentType("text/html; charset=EUC-KR");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
            _jspx_out = out;

            out.write("<html>\n");
            out.write("<body>\n");
            out.write("\t<H2>Counter test</H2>\n");
            out.write("\t");
            int count=0;
            out.write("\n");
            out.write("\t<P>\n");
            out.write("\t\t당신은 ");
            out.println(++count);
            out.write("\t\t번째 접속자입니다.\n");
            out.write("\t</P>\n");
            out.write("\n");
            out.write("</body>\n");
            out.write("</html>");
        } catch (Throwable t) {
            ...
        }
    }
}
```

선언 (Declaration)

- 생성되는 servlet 내에서 사용되는 method나 member variable 선언
- `<%! declare... %>`
- Servlet 클래스의 내부에, 다른 method의 외부에 위치

```
<H2>Counter test</H2>
<%! int count=0; %>
<P>
당신은 <% out.println(++count); %>번째 접속자입니다.
</P>
```

SimpleCounter.jsp

생성된 servlet

```
public final class SimpleCounter_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    int count=0;

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

        ...
    }
}
```

Counter JSP 만들기

- 여러 페이지 통합 카운터

SingletonCounter.java

```
package myapp.ch06;  
  
public class SingletonCounter {  
    private static int count = 0;  
    public static int getCount() {  
        return count;  
    }  
    public static void addCount() {  
        count++;  
    }  
}
```

Counter 1

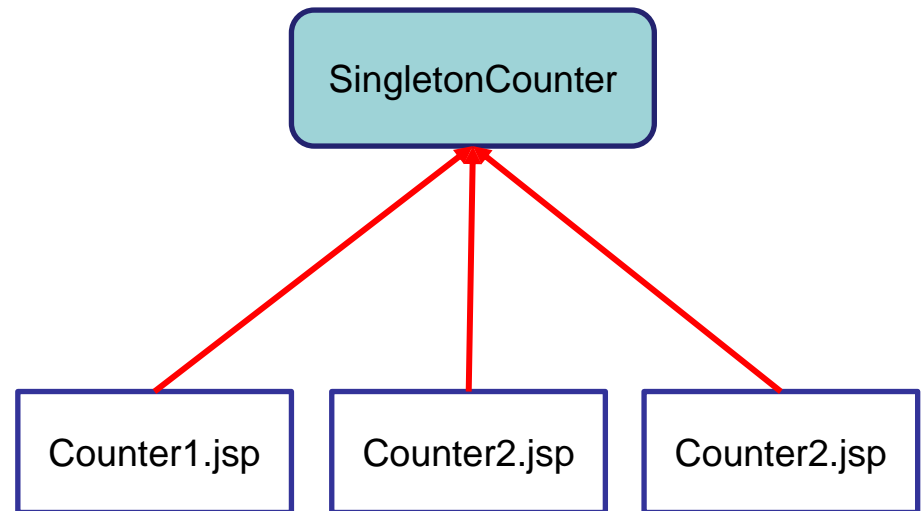
당신은 1 번째 사용자입니다.

Counter 2

당신은 2 번째 사용자입니다.

설계 원칙

JSP 안에 복잡한 method와 로직을 넣지 말고,
로직을 담당하는 부분을 별도로 구현하자!



```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="myapp.ch06.SingletonCounter" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Counter 1</title>
</head>
<body>
    <h2>Counter 1</h2>
    <% SingletonCounter.addCount(); %>
    <p> 당신은 <% out.println(SingletonCounter.getCount()); %>번째 사용자입니다. </p>
</body>
</html>

```

Counter1.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="myapp.ch06.SingletonCounter" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Counter 2</title>
</head>
<body>
    <h2>Counter 2</h2>
    <% SingletonCounter.addCount(); %>
    <p> 당신은 <% out.println(SingletonCounter.getCount()); %>번째 사용자입니다. </p>
</body>
</html>

```

Counter2.jsp

Counter JSP 만들기

```
package org.apache.jsp.ch06;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import jspbook.ch06.Counter;

public final class CounterJSP_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    private static java.util.List _jspx_dependants;

    public Object getDependants() {
        return _jspx_dependants;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;
```

Counter1_jsp.java

Servlet 지역변수 선언


```

try {
    _jspxFactory = JspFactory.getDefaultFactory();
    response.setContentType("text/html; charset=EUC-KR");
    pageContext = _jspxFactory.getPageContext(this, request, response,
        null, true, 8192, true);
    _jspx_page_context = pageContext;
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    _jspx_out = out;

    out.write("WrWn");
    out.write("Wn");
    out.write("<!DOCTYPE html PUBLIC W-//W3C//DTD HTML 4.01 Transitional//ENW W'http://www.w3.org/TR/html4/loose.dtdW'>Wn");
    out.write("<html>Wn");
    out.write("<head>Wn");
    out.write("<meta http-equiv=W'Content-TypeW' content=W'text/html; charset=EUC-KRW'>Wn");
    out.write("<title>Counter JSP</title>Wn");
    out.write("</head>Wn");
    out.write("<body>WrWn");
    out.write("Wt<H2>Counter test</H2>Wn");
    out.write("W");
    Counter.addCount();
    out.write("WrWn");
    out.write("Wt<P>WrWn");
    out.write("W달식은 ");
    out.println(Counter.getCount());
    out.write("번째 접속자입니다.WWn");
    out.write("Wt</P>Wn");
    out.write("</body>Wn");
    out.write("</html>");
} catch (Throwable t) {
    if (!(t instanceof SkipPageException)){
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0)
            out.clearBuffer();
        if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
    }
} finally {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
}

```

내장 객체 초기화

결과물 출력

exception 처리

표현식 (Expression)

- out.println()으로 표현된 scriptlet을 대체하는 간단한 방식
- <%= *expression* %>
- 표현식의 내용이 out.println()의 매개변수가 됨

당신은 <%= out.println(Counter.getCount()); %> 번째 접속자입니다.

||

<%= Counter.getCount() %>

; 이 안붙도록 주의!

JSP에서는 되도록이면 out.println()을 사용하지 말자!

JSP 내장 객체

- JSP로부터 자동 생성되는 **servlet** 코드 내에 이미 선언/할당되는 객체
- **Scriptlet** 내에서 다른 변수와 마찬가지로 사용

```
JspFactory _jspxFactory = null;  
PageContext pageContext = null;  
HttpSession session = null;  
ServletContext application = null;  
ServletConfig config = null;  
JspWriter out = null;  
Object page = this;  
JspWriter _jspx_out = null;  
PageContext _jspx_page_context = null;
```

Servlet 지역변수 선언

```
try {  
    _jspxFactory = JspFactory.getDefaultFactory();  
    response.setContentType("text/html; charset=EUC-KR");  
    pageContext = _jspxFactory.getPageContext(this, request, response,  
        null, true, 8192, true);  
    _jspx_page_context = pageContext;  
    application = pageContext.getServletContext();  
    config = pageContext.getServletConfig();  
    session = pageContext.getSession();  
    out = pageContext.getOut();  
    _jspx_out = out;
```

내장 객체 초기화

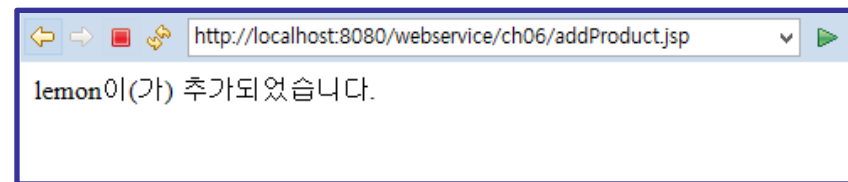
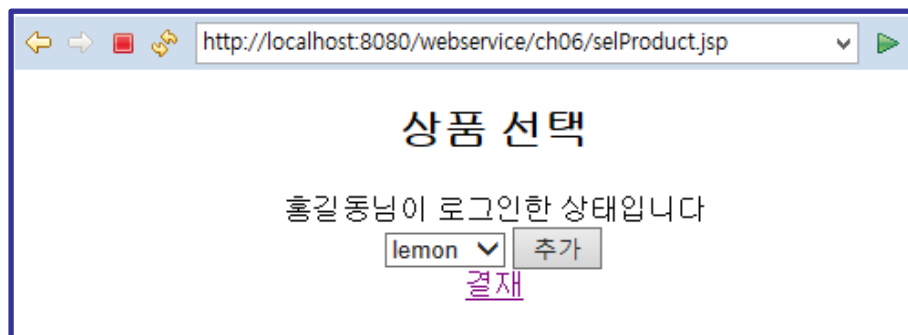
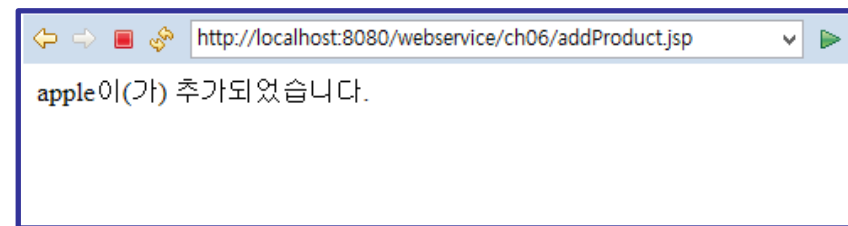
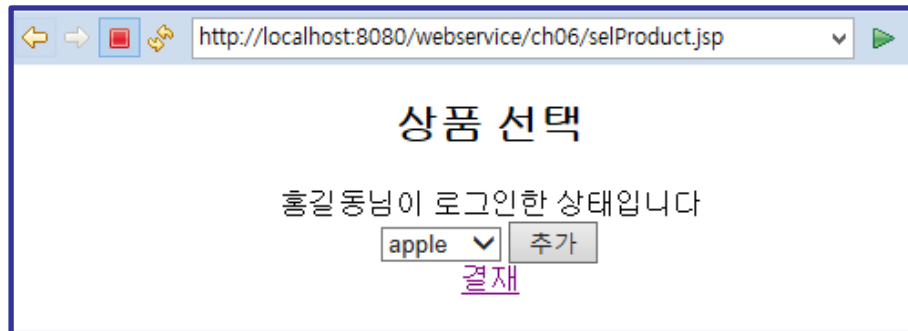
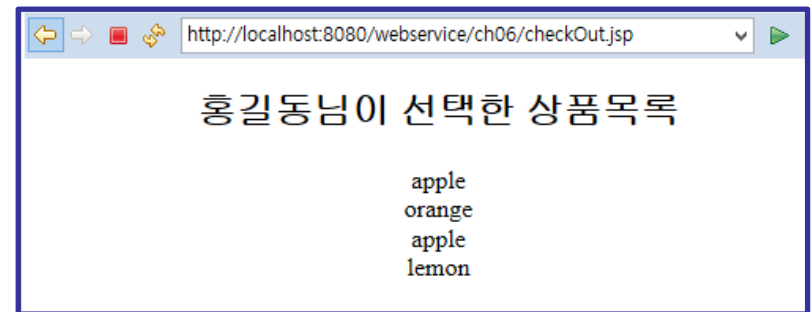
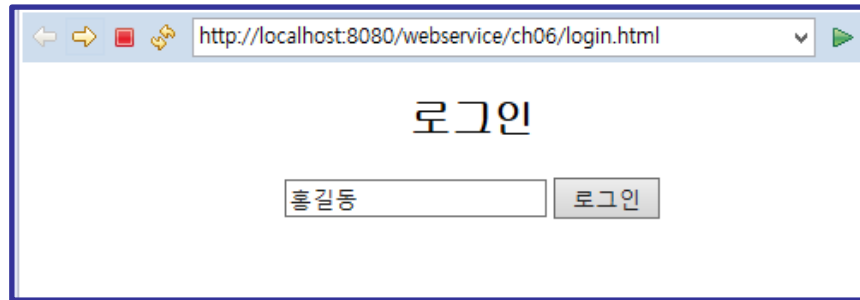
```
<P>  
당신은 <% out.println(Counter.getCount()); %>번째 접속자입니다.  
</P>
```

JSP 내장 객체

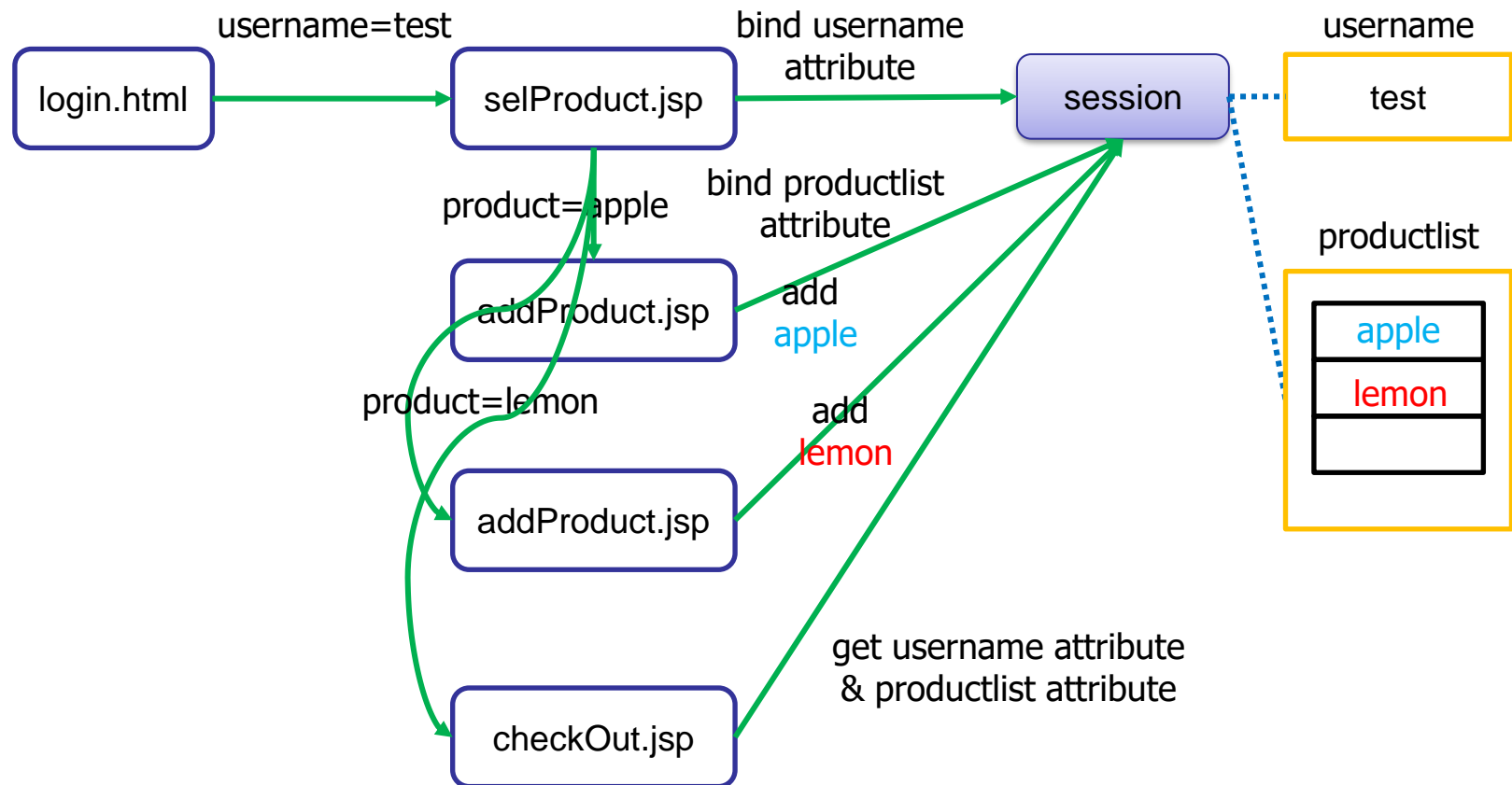
response.getWriter()를 통해
얻는 PrintWriter와 유사

Servlet 객체	JSP 내장 객체 instance명
HttpServletRequest	request
HttpServletResponse	response
JspWriter	out
ServletContext	application
ServletConfig	config
HttpSession	session
PageContext	pageContext
JspException	exception
Object	page

JSP를 이용한 장바구니



JSP를 이용한 장바구니



JSP를 이용한 장바구니

- login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>
    body {text-align:center;}
</style>
</head>
<body>
    <h2>로그인</h2>
    <form method=post action=selProduct.jsp>
        <input type=text name=username />
        <input type=submit value=로그인 />
    </form>
</body>
</html>
```

JSP를 이용한 장바구니

- selProduct.jsp

```
<%@ 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>
<style>
    body {text-align: center;}
</style>
</head>
<%
    request.setCharacterEncoding("UTF-8");
    String name = request.getParameter("username");
    session.setAttribute("username", name);
%>
<body>
    <h2>상품 선택</h2>
    <%= name %> 님이 로그인한 상태입니다
    <form method=post action=addProduct.jsp>
        <select name=product>
            <option>apple</option>
            <option>orange</option>
            <option>lemon</option>
        </select>
        <input type=submit value="추가" />
    </form>
    <a href=checkout.jsp>계산</a>
</body>
</html>
```

한글 parameter가
깨지는 것을 방지

JSP를 이용한 장바구니

- addProduct.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="java.util.ArrayList" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    request.setCharacterEncoding("UTF-8");
    String product = request.getParameter("product");
    ArrayList<String> list = (ArrayList<String>)session.getAttribute("productlist");
    if (list == null) list = new ArrayList<String>();
    list.add(product);
    session.setAttribute("productlist", list);
    out.println(product + "이(가) 추가되었습니다.");
%>
</body>
</html>
```

JSP를 이용한 장바구니

- checkOut.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="java.util.ArrayList"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
    body {text-align: center;}
</style>
<title>Insert title here</title>
</head>
<% request.setCharacterEncoding("UTF-8"); %>
<body>
    <h2> <%=session.getAttribute("username") %>님이 선택한 상품목록</h2>
    <%
        ArrayList<?> list = (ArrayList<?>)session.getAttribute("productlist");
        if (list == null) out.println("선택한 상품이 없습니다!");
        else
            for (Object product:list)
                out.println(product + "<br>");
    %>
</body>
</html>
```

Action

- JSP action의 기능
 - JSP 페이지간의 흐름 제어
 - Java applet, Java Beans 컴포넌트 지원
- Action의 종류
 - include action
 - forward action
 - plugin action
 - useBean action

include action

- 현재의 페이지에 다른 페이지를 포함시키는 기능 제공
- 사용법
 - `<jsp:include page="JSP name" />`
- include directive vs. include action
 - include directive
 - 두 파일 내용을 하나로 합친 후 컴파일
 - 정적인 페이지를 포함시킬 때 사용
 - include action
 - 실행 시점에 해당 파일을 호출해서 그 결과를 현 페이지에 포함
 - 동적인 페이지를 포함시킬 때 사용

include action

include_action.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <H2>include_action.jsp에서 footer.jsp 호출</H2>
    include_action.jsp에서 출력한 메시지
    <HR>
    <jsp:include page="footer.jsp">
        <jsp:param name="email" value="admin@myserver.com" />
        <jsp:param name="tel" value="123-4567" />
    </jsp:include>
    <HR>
    include_action.jsp에서 출력한 메시지
</body>
</html>
```

include action

footer.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    footer.jsp에서 출력한 메시지 <BR>
    Email : <%= request.getParameter("email") %> <BR>
    Tel : <%= request.getParameter("tel") %>
</body>
</html>
```

http://localhost:8080/jspbook/ch06/include_action.jsp

include_action.jsp에서 footer.jsp 호출

include_action.jsp에서 출력한 메시지

footer.jsp에서 출력한 메시지
Email : admin@myserver.com
Tel : 123-4567

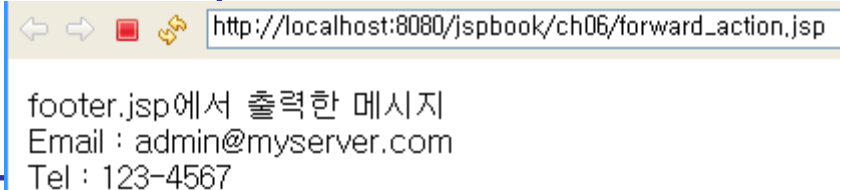
include_action.jsp에서 출력한 메시지

forward action

- 현재 페이지에서 제어권을 다른 페이지로 넘길 때 사용
- Servlet에서 `RequestDispatcher.forward()`와 동일한 기능
- 사용법
 - `<jsp:forward page="JSP name" />`

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <H2>forward_action.jsp에서 footer.jsp 호출</H2>
    forward_action.jsp에서 출력한 메시지
    <HR>
    <jsp:forward page="footer.jsp">
        <jsp:param name="email" value="admin@myserver.com" />
        <jsp:param name="tel" value="123-4567" />
    </jsp:forward>
    <HR>
    forward_action.jsp에서 출력한 메시지
</body>
</html>
```

forward_action.jsp

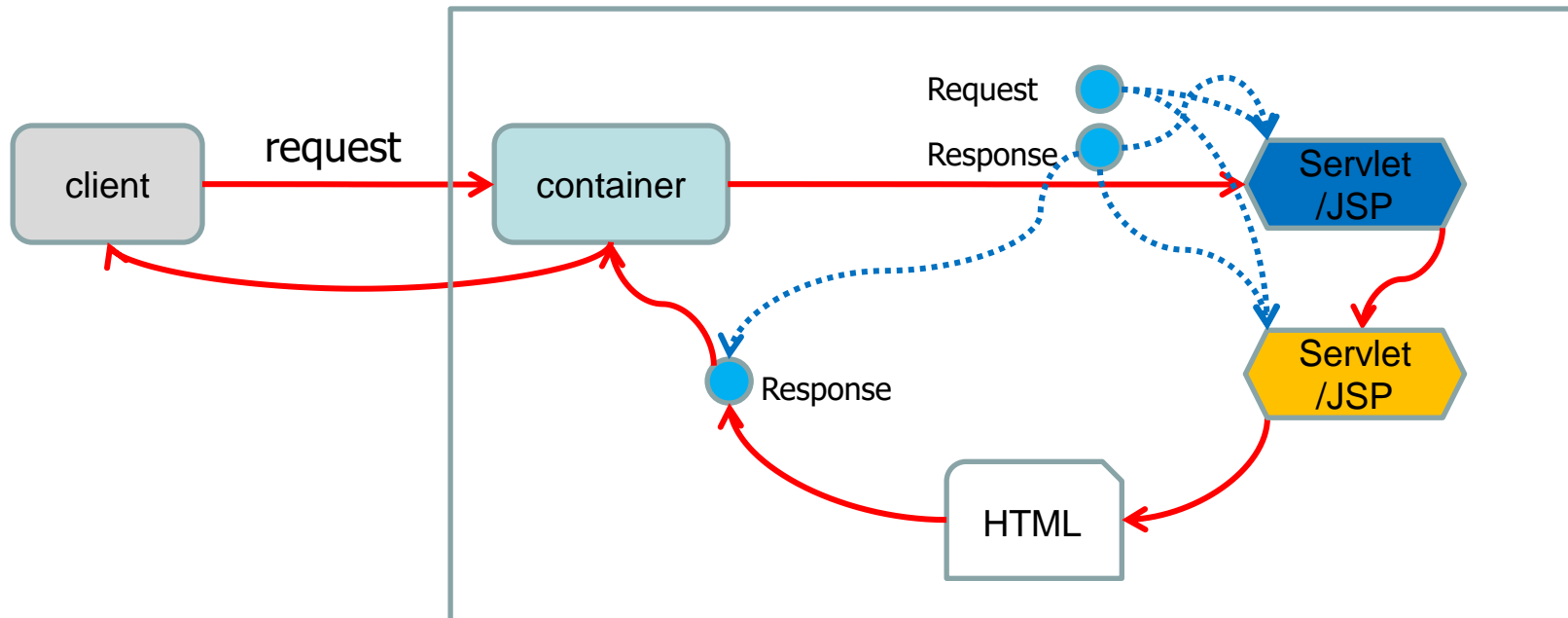


http://localhost:8080/jspbook/ch06/forward_action.jsp

footer.jsp에서 출력한 메시지
Email : admin@myserver.com
Tel : 123-4567

forward vs. redirect

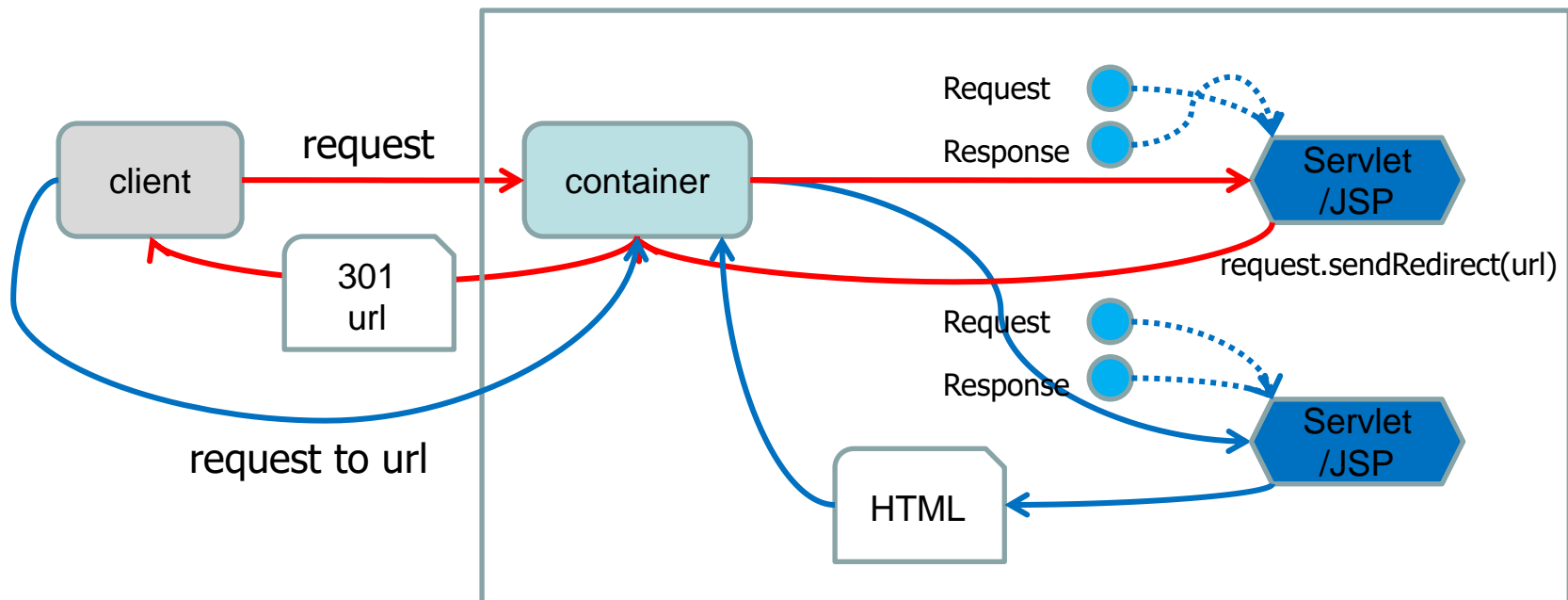
- forward or request dispatch
 - 요청에 대한 처리를 서버 내부에서 다른 component에게 위임
 - 요청에 관련된 request, response 객체가 함께 전달되어 처리
 - In servlet : RequestDispatcher 사용
 - In JSP : forward action 사용



forward vs. redirect

- redirect

- 요청에 대해 새로운 URL을 client에게 전송해서 Client가 새로운 URL로 다시 요청을 보내도록 함
- 새 URL에 대한 요청이기 때문에 새로운 request, response 객체가 생성
- In servlet & JSP : response 객체의 sendRedirect() 이용



forward vs. redirect

page_control.jsp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
  <H2>forward, sendRedirect test</H2>
  <hr>
  <form method=post action="forward_action2.jsp">
    forward action: <input type=text name=username>
    <input type=submit value="submit">
  </form>
  <form method=post action="response_sendRedirect.jsp">
    response.sendRedirect: <input type=text name=username>
    <input type=submit value="submit">
  </form>
</body>
</html>
```

← → 🚫 🔍 http://localhost:8080/jspbook/ch06/page_control.jsp

forward, sendRedirect test

forward action:

response.sendRedirect:

forward vs. redirect

forward_action2.jsp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <% request.setCharacterEncoding("euc-kr"); %>
    <jsp:forward page="page_control_end.jsp">
        <jsp:param name="tel" value="123-4567" />
    </jsp:forward>
</body>
</html>
```

http://localhost:8080/jspbook/ch06/page_control.jsp

forward, sendRedirect test

forward action:

response.sendRedirect:

page_control_end.jsp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    page_control_end.jsp output page
    <hr>
    name : <%= request.getParameter("username") %> <br>
    tel  : <%= request.getParameter("tel") %>
</body>
</html>
```

http://localhost:8080/jspbook/ch06/forward_action2.jsp

page_control_end.jsp output page

name : test
tel : 123-4567

forward vs. redirect

response.sendRedirect.jsp

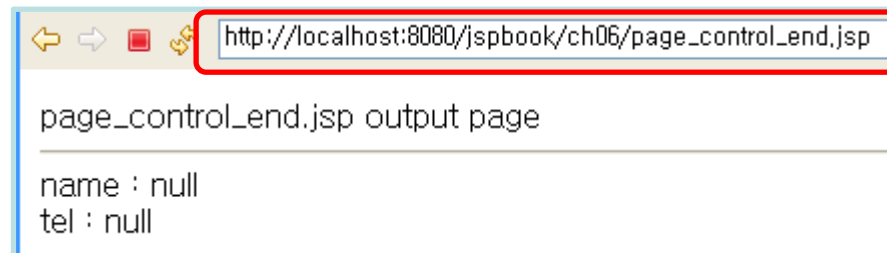
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <% response.sendRedirect("page_control_end.jsp"); %>
</body>
</html>
```



forward, sendRedirect test

forward action: submit

response.sendRedirect: submit



page_control_end.jsp output page

name : null
tel : null