

JSP

JSP 왜 필요한가?

기존 Servlet 구성

```
@WebServlet("/requestjspServlet")
public class requestjspServlet extends HttpServlet {

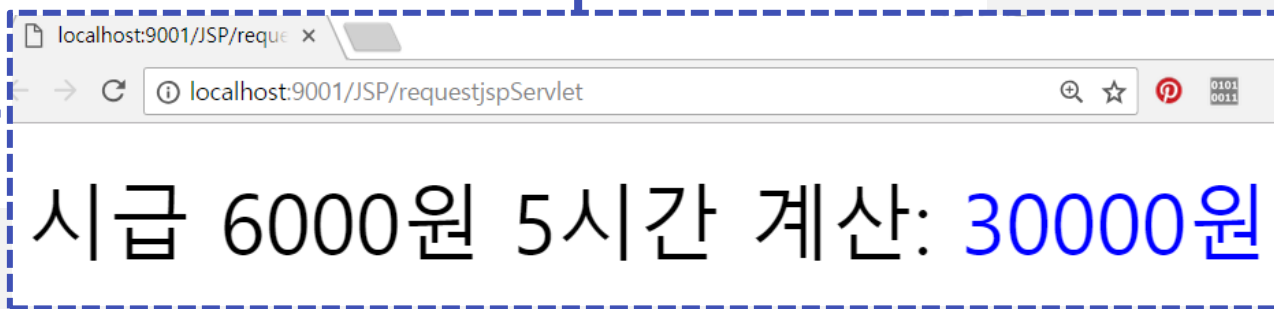
    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

        response.setContentType("text/html; charset=euc-kr");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<title></title>");
        out.println("<body>");
        out.println("<p>시급 6000원 5시간 계산 : ");
        out.println("<font color='blue'>" + (6000*5) + "원 </font>");
        out.println("</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

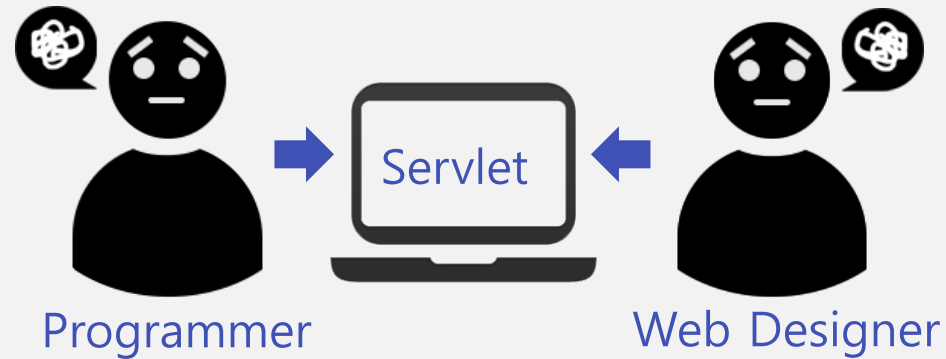
Business Logic

Presentation Logic



JSP 왜 필요한가?

1. Servlet은 수정이 어렵다.



2. 프로그램이 복잡해지면 Servlet을 통해 출력되는 HTML페이지를 상상하기 어렵다.

JSP 왜 필요한가?

Servlet -> Java기반

```
import java.io.IOException;

@WebServlet("/requestjspServlet")
public class requestjspServlet extends HttpServlet {
    protected void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<title>시급계산기</title>");
        out.println("<body>");
        out.println("<p>시급 6000원 5시간 계산 : <font color='blue'>" + (6000*5) + "원</font></p>");
        out.println("</body>");
        out.println("</html>");

    }
}
```

HTML출력

Java코드

JSP -> HTML기반

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>시급계산기</title>
</head>
<body>
    <p> 시급 6000원 5시간 계산 : <font color='blue'><% out.println(6000 * 5); %>원 </font></p>
</body>
</html>
```

Java코드

JSP vs Servlet

	장점	단점	형식
서블릿	복잡한 로직 구현에 적합함	JAVA코드 중심 화면작성 작업과 수정이 어려움	<div>Java 코드</div> <div>HTML 코드</div>
JSP	HTML코드 중심 → 화면작성 작업, 수정용이	소스보기로 소스가 공개되어 중요정보에 부적합	<div>HTML 코드</div> <div>Java 코드</div>

→ 로직 구현은 Servlet 클래스에, HTML중심의 코드는 JSP에

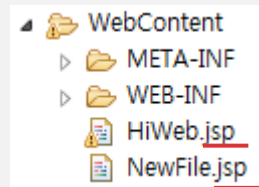
JSP란?

JSP(Java Server Page)

HTML내에 Java 코드를 작성하고 웹 서버에서 웹 페이지를 생성하여

Web browserd에 돌려주는 Java를 기반으로 한 스크립트 언어.

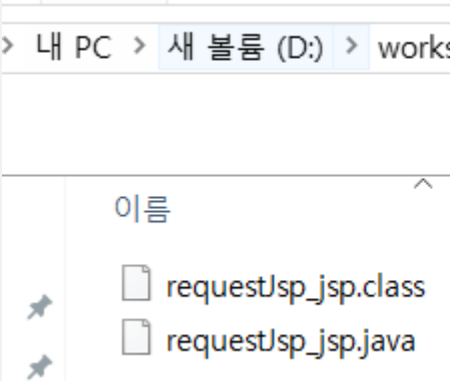
-jsp 확장자



-동적으로 작동하여 응답은 html을 이용함
(동적 웹 애플리케이션 컴포넌트)

-jsp->서블릿->클래스->html으로 변환되어 실행됨

JSP Life Cycle

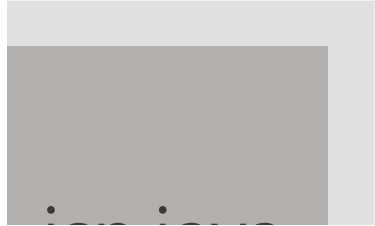


```
public void _jspService(final javax.servlet.http.HttpServletRequest request, final javax.servlet.http.HttpServletResponse response)
    throws java.io.IOException, javax.servlet.ServletException {

    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;

    try {
        response.setContentType("text/html; charset=UTF-8");
        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("\r\n");
        out.write("<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"\r\n");
        out.write("<html\r\n");
        out.write("<head\r\n");
        out.write("<meta http-equiv='Content-Type' content='text/html; charset=UTF-8'\r\n");
        out.write("<title>시급 계산기</title\r\n");
        out.write("</head\r\n");
        out.write("<body\r\n");
        out.write("<p>시급 6000원 5시간 계산 : <font color='blue'>");
        out.println(6000 * 5);
        out.write("원\r\n");
        out.write("</body\r\n");
        out.write("</html\r\n");
        out.write("\r\n");
        out.write("\r\n");
    } catch (java.lang.Throwable t) {
        if (!(t instanceof javax.servlet.jsp.SkipPageException)){
            out = _jspx_out;
            if (out != null && out.getBufferSize() != 0)
                try {
                    if (response.isCommitted()) {
                        out.flush();
                    } else {
                        out.clearBuffer();
                    }
                }
            }
        }
    }
}
```



ory 적재
실행

JSP 구성요 소

```
<%@ 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>
    <%!
        public int sum(int num1, int num2){
            return num1 + num2;
        }
    %>

    <%
        int num1 = 20;
        int num2 = 10;
        int sum = num1 + num2;
    %>

    <%=sum %>
</body>
</html>
```

지시자
(Directive)

선언문(Declaration)

스크립트릿
(Scriptlet)
표현식(Expression)

Scriptlet(스크립트릿)

1. 스크립트릿 <% %>

- JSP문서 안에 JAVA코드를 넣기 위해 사용함.
그러나 JSP는 모든 것을 태그화하여 쓰는 것이 목적이므로 줄여가도록 함.

```
<body>
```

```
<%
```

```
    int num1 = 20;
```

```
    int num2 = 10;
```

```
    int sum = num1 + num2;
```

```
%>
```

```
</body>
```

Expression(표현식)

2. 표현식

<%= %> Web browser에 결과값을 출력하기 위해 사용

```
<%=sum %>
```

```
<%=sum+100 %>
```

```
<%= Math.sqrt(sum) %>
```

상수나 변수

연산자

메소드호출

JSP 실습(1)

ex02hap100.jsp을 만들고
1~100까지의 합을 구하시오.

ex03table.jsp을 만들고
아래와 같은 table에 10개의 방을 만드시오.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Declartion(선언문)

3. 선언문

<%! 선언부에는 변수 선언이나, 메서드를 선언하여 사용

- 이 변수와 메서드는 JSP 페이지로부터 변환된 서블릿 클래스의 멤버로 변환

```
<%! final static int MAX=10000; %>
```

변수 선언

```
<%!  
    private int add(int num1, int num2) {  
        int sum = num1 + num2;  
        return sum;  
    }  
%>
```

메소드 선언

```
package org.apache.jsp;  
  
import javax.servlet.*;  
import javax.servlet.http.*;  
import javax.servlet.jsp.*;  
  
public final class declartion_jsp extends org.apache.jasper.runtime.HttpJspBase  
    implements org.apache.jasper.runtime.JspSourceDependent {  
  
    final static int MAX = 10000;  
  
    private int add(int num1, int num2){  
  
        int sum = num1+num2;  
        return sum;  
    }  
  
    private static final javax.servlet.jsp.JspFactory _jspxFactory =  
        javax.servlet.jsp.JspFactory.getDefaultFactory();
```

declartion_jsp.jav

**+, - , *, / 값을 return하는 메소드를 선언문으로
설계하고
해당 메소드를 호출하시오**

```
2+2 : 4  
2-2 : 0  
2*2 : 4  
2/2 : 1
```

ex04cal.jsp

JSP 실습(3)

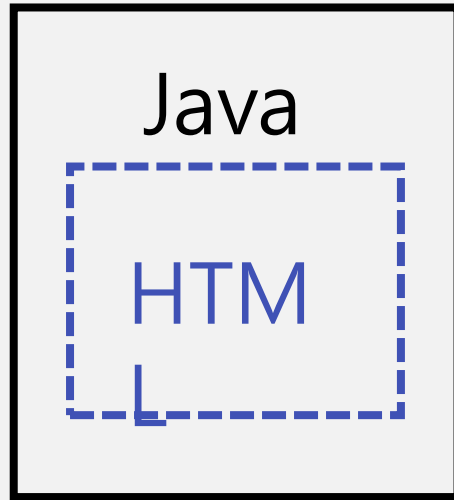
NUMBER1	<input type="text" value="3"/>
NUMBER2	<input type="text" value="3"/>
<input type="button" value="입력"/>	

ex05makecal.ht
ml

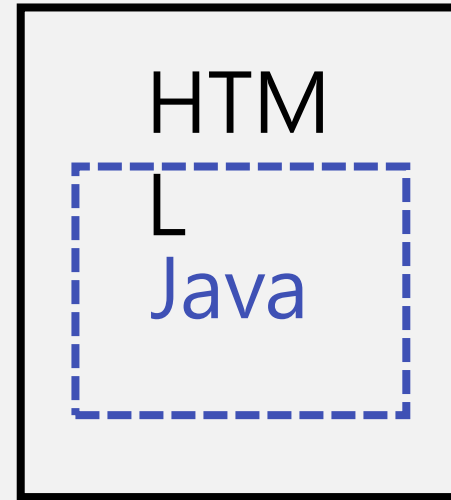
3+3=6
3-3=0
3*3=9
3/3=1

ex05makecal.js
p

Review



복잡한
로직
Servlet



화면 구
성
JSP

JSP 구성요

소

1.

`<%@ %>`

2.
3. `<%! %>`

4. `<%= %`

`>`

표현

선언

문서

자크립트

릿

지시자(Directive)

1. 지시자

<%@ %>
- Web Container가 JSP page를 Servlet class로 변환할 때
필요한 정보를 기술하기 위해 사용

page 지시자

taglib 지시
자

include 지시
자

지시자(Directive)

4-1. page 지시자

JSP페이지의 전체적인 환경설정을 할 때 사용

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>

<%
    ArrayList<String> arr = new ArrayList<String>();

    arr.add("지시자");
%>

<%= arr.get(0) %>

</body>
</html>
```

페이지의 전체적인 속성
을 지정



```
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>

<%
    ArrayList<String> arr = new ArrayList<String>();

    arr.add("지시자");
%>

<%= arr.get(0) %>

</body>
</html>
```

import문 사
용

지시자(Directive)

속성	설명	사용빈도	기본값
contentType	브라우저로 내보내는 내용의 MIME 형식 지정 및 문자집합 지정	매우 높음	text/html;charset=ISO-8859-1
pageEncoding	현재 JSP 페이지에 대한 문자집합 지정	거의 사용하지 않음	ISO-8859-1
import	현재 JSP 페이지에서 사용할 Java 패키지나 클래스를 지정	높음	
errorPage	에러가 발생할 때에 대신 호출되어 처리될 JSP 페이지 지정	중간	
isErrorPage	현재 JSP 페이지가 에러를 핸들링하는 페이지인지 지정	중간	false
info	현재 JSP 페이지에 대한 설명	거의 사용하지 않음	
buffer	버퍼의 크기	거의 사용하지 않음	8KB
autoflush	버퍼의 내용을 자동으로 브라우저로 보낼지에 대한 설정	거의 사용하지 않음	true
session	세션의 사용유무 설정	중간	true
language	스크립트 언어 종류 지정	거의 사용하지 않음	java
isThreadSafe	현재 JSP 페이지에 대한 접근은 한 순간에 하나의 스레드만 접근하도록 하여 동시 접근으로 인한 위험성 제거 설정	거의 사용하지 않음	true
extends	현재 JSP 페이지를 기본적인 클래스가 아닌 다른 클래스로 부터 상속하도록 변경	거의 사용하지 않음	javax.servlet.jsp.HttpJspPage

지시자(Directive)

JSP 페이지가 생성하는 문서의 종류와 인코딩 방식을 지정하기 위해 사용

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

한글이 포함된
JSP
페이지일 경우

```
<%@page contentType= "text/html " %>
```

영문자로만
구성된 JSP 페이지
일 경우

지시자(Directive)

JSP 페이지가 서블릿 클래스로 변환될 때 contentType Attribute의 값은 setContentType 메서드 호출문의 파라미터가 됨

한글 코드 지시자가 있을 때

```
try {  
    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();  
}
```

지시자(Directive)

Java의 **import** 문과 마찬가지로 다른 패키지에 속하
는
클래스나 인터페이스를 **import**하는 역할

```
<%@page import= "java.util.GregorianCalendar" %>
```

→ java.util 패키지의
GregorianCalendar 클래스를
import

**ex06calendar.jsp를 만들고 현재
시간을 출력하는 페이지를 만들어
보자.**

지시자(Directive)

0과 2를 나워볼까요?

$\langle \% = 2 / 0 \% \rangle$

0을 나누는 것은 불
가능

지시자(Directive)

```
HTTP Status 500 - An exception occurred processing JSP page /NewFile.jsp at line 15

type Exception report
message An exception occurred processing JSP page /NewFile.jsp at line 15
description The server encountered an internal error that prevented it from fulfilling this request.
exception
org.apache.jasper.JasperException: An exception occurred processing JSP page /NewFile.jsp at line 15

12: </head>
13: <body>
14: 0과 2를 나눠볼까요?
15: <%=2/0%>
16: </body>
17: </html>

Stacktrace:
org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:574)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:476)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:396)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:340)
javax.servlet.http.HttpServlet.service(HttpServlet.java:729)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)

root cause
java.lang.ArithmeticException: / by zero
org.apache.jsp.NewFile_jsp._jspService(NewFile_jsp.java:111)
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
javax.servlet.http.HttpServlet.service(HttpServlet.java:729)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:438)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:396)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:340)
javax.servlet.http.HttpServlet.service(HttpServlet.java:729)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)

note The full stack trace of the root cause is available in the Apache Tomcat/8.0.23 logs.

Apache Tomcat/8.0.23
```



죄송합니다!

서버 오류를 점검 중입니다.
잠시만 기다려 주십시오.
문의전화) 000-0000-0000

errorpage.jsp

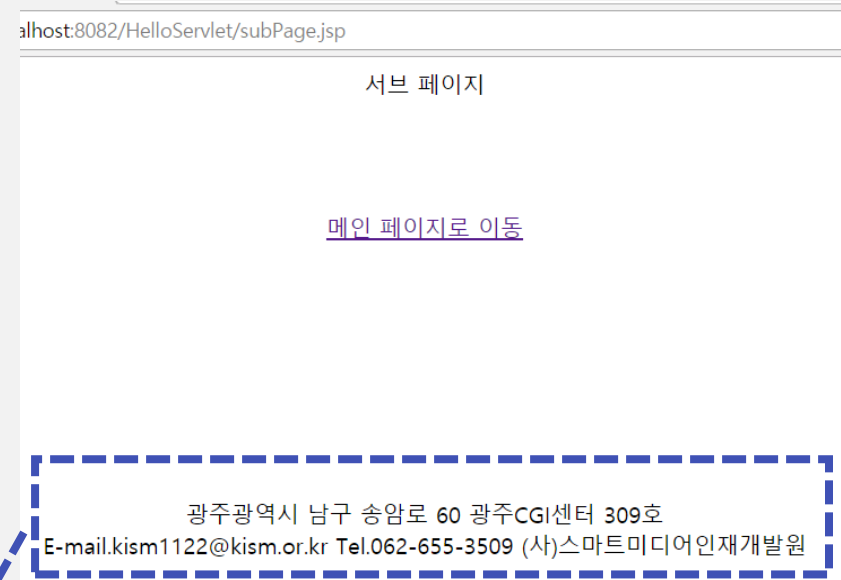
지시자(Directive)

4-2. include 지시자

현재 페이지에 다른 파일의 내용을 삽입할 때



ex07mainPage.jsp



ex07subPage.jsp

ex06calendar.jsp

지시자(Directive)

4-3. taglib 지시자

자 태그 라이브러리에서 태그를 가져와 사용할 수 있는 기능 제공 ex) JSTL

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!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>
    <center>
        <c:choose>
            <c:when test="${empty applicationScope.visit }"> <!-- null 대신 empty를 앞에 써준다 -->
                <c:set var="visit" value="1" scope="application"></c:set>
            </c:when>
            <c:otherwise>
                <c:set var="visit" value="${visit+1 }" scope="application"></c:set>
            </c:otherwise>
        </c:choose>
        ${applicationScope.visit}방문자 수입니다.
    </center>
    <h1>로그인을 해주세요.</h1>
    <form action="" method="">
        ID<input type="text" name="id"><br>
        PS<input type="text" name="ps"><br>
        <input type="submit" value="로그인"><br>
        <hr>
        회원이 아니시라면 <a href="JoinView.jsp">회원가입</a><br>
    </form>
    </center>
</body>
</html>
```

Annotation(주석)

5. 주석 <%-- --

%> 프로그램 설명, 테스트 목적으로 사용되는 태그

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
1
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <title>Insert title here</title>
7 </head>
8 <body>
9
10     <!-- HTML 주석 -->
11
12
13
14
15 </body>
16 </html>
</body>
</html>
```

내장객체(Implicit Object)

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

```
    final javax.servlet.jsp.PageContext pageContext;  
    javax.servlet.http.HttpSession session = null;  
    final javax.servlet.ServletContext application;  
    final javax.servlet.ServletConfig config;  
    javax.servlet.jsp.JspWriter out = null;  
    final java.lang.Object page = this;  
    javax.servlet.jsp.JspWriter _jspx_out = null;  
    javax.servlet.jsp.PageContext _jspx_page_context = null;
```

JSP page 안에 선언을 하지 않고도 사용할 수
있는 객체

내장객체(Implicit Object)

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8" %>
<!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">
<HEAD><TITLE>정수를 순서대로</TITLE></HEAD>
    <BODY>
        <H3>정수를 순서대로</H3>
        <%
            String str = request.getParameter( "MAX" );
            int max = Integer.parseInt(str);
            for (int cnt = 1; cnt <= max; cnt++)
                out.println(cnt + "<BR>");
        %>
    </body>
</html>
```

내장객체(Implicit Object)

JSP page안에서 내장객체를 사용할 수 있는 이유는 Web container
가

JSP page를 Servlet class로 변환시키면서 자동으로 내장 객체를 선언

변수 이름	제공하는 기능/변수의 역할	변수 타입
request	doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletRequest
response	doGet, doPost 메서드의 두 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletResponse
out	웹 브라우저로 HTML 코드를 출력하는 기능	javax.servlet.jsp.JspWriter
application	JSP 페이지가 속하는 웹 애플리케이션에 관련된 기능	javax.servlet.ServletContext
config	JSP 페이지의 구성 정보를 가져오는 기능	javax.servlet.ServletConfig
pageContext	JSP 페이지 범위 내에서 사용할 수 있는 데이터 저장 기능 등	javax.servlet.jsp.PageContext
Session	세션에 관련된 기능	javax.servlet.http.HttpSession
Page	JSP 페이지로부터 생성된 서블릿	java.lang.Object
Exception	익셉션 객체	java.lang.Throwable

out 객체

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

```
    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;
```

```
    out.write("\r\n");
    out.write("<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">\r\n");
    out.write("<html>\r\n");
    out.write("<head>\r\n");
    out.write("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\"");
    out.write(">\r\n");
    out.write("<title>시급계산기</title>\r\n");
    out.write("</head>\r\n");
    out.write("<body>\r\n");
    out.write("\t<p>시급 6000원 5시간 계산 : <font color='blue'>");
    out.write("111 (6000 * 5)");
```

호출

출력

out 객체

```
10개의 방 출력
<table border="1">
<tr>
<%
    for(int i=0;i<10;i++){
        %>
        <td><%=i%></td>
        <%
    }
    %>
</tr>
</table>
```

```
<table border="1">
<tr>
<%
    for(int i=0;i<10;i++){
        out.print("<td>"+i+"</td>");
    }
    %>
</tr>
</table>
```

ex08out.jsp

Scriptlet을 다중으로 사용하지 않음으로써
코드가 간결해짐

out 객체

```
<%=year %>년 <%=month+1 %>월 <%=date %>일<br>
<%=hour %>시 <%=min %>분 <%=sec %>초
```

<표현식을 사용하는 경우>

```
<%=if (num%2==0) { %>
    짝수입니다.
<%=} else { %>
    홀수입니다.
<%=} %>
```

```
<%=
    if (num%2==0) {
        out.print("짝수입니다.");
    } else {
        out.print("홀수입니다.");
    }
%>
```

<out객체를 사용하는 경우>

Client가 Server로 데이터를 전달할 때
HTTP Message로 구성해 여러 정보를 사용할 수
있도록
저장하고 관리하는 내장객체

request 객체

학점확인프로그램

이름	<input type="text" value="차현석"/>
JAVA점수	<input type="text" value="100"/>
WEB점수	<input type="text" value="100"/>
IOT점수	<input type="text" value="100"/>
ANDROID점수	<input type="text" value="100"/>
<input type="button" value="확인하기!"/>	

ex09request.jsp

입력한 데이터들을 아래의 jsp 파일에 전송할 수 있는 조건을 완성하시오.

파일 이름 : ex09request.jsp

전송방식 : post

request객체

학점확인프로그램	
이름	차현석
JAVA점수	100
WEB점수	100
IOT점수	100
ANDROID점수	100
평균	100.0
학점	A+ 강조태그적용

ex09requestResult.jsp

평균과 학점을 구하시오

평균 : 100~95 **A+** / 94~85 **A** / 84~80 **B+** / 79~70 **C** / ~ **F**

몇 개의 층을 만드실건가요

층

몇 개의 방을 만드실건가요

방

ex10maketable.html

5층

5방

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4

ex10maketable.jsp

request객체

랜덤당첨게임

랜덤숫자입력

3

생성하기

ex1createInput.html

랜덤당첨점 작성

주제: 오늘 점심 메뉴는?

아이템1: 짜장면

아이템2: 김밥나라

아이템3: 한솔

시작

ex1createInput.jsp

랜덤당첨결과

오늘 점심 메뉴는?

짜장면

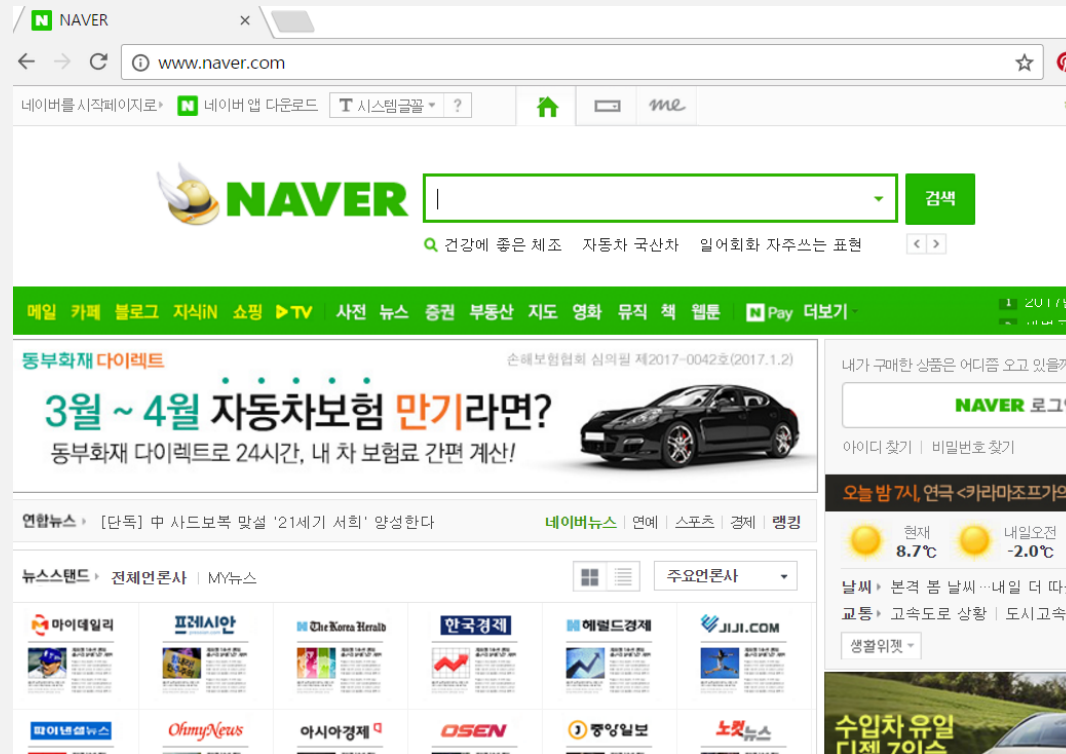
ex1randomWinner.jsp

response객
체

JSP page의 실행결과를 Web Browser로
되돌려줄 때 사용하는 내장객체

response.sendRedirect()

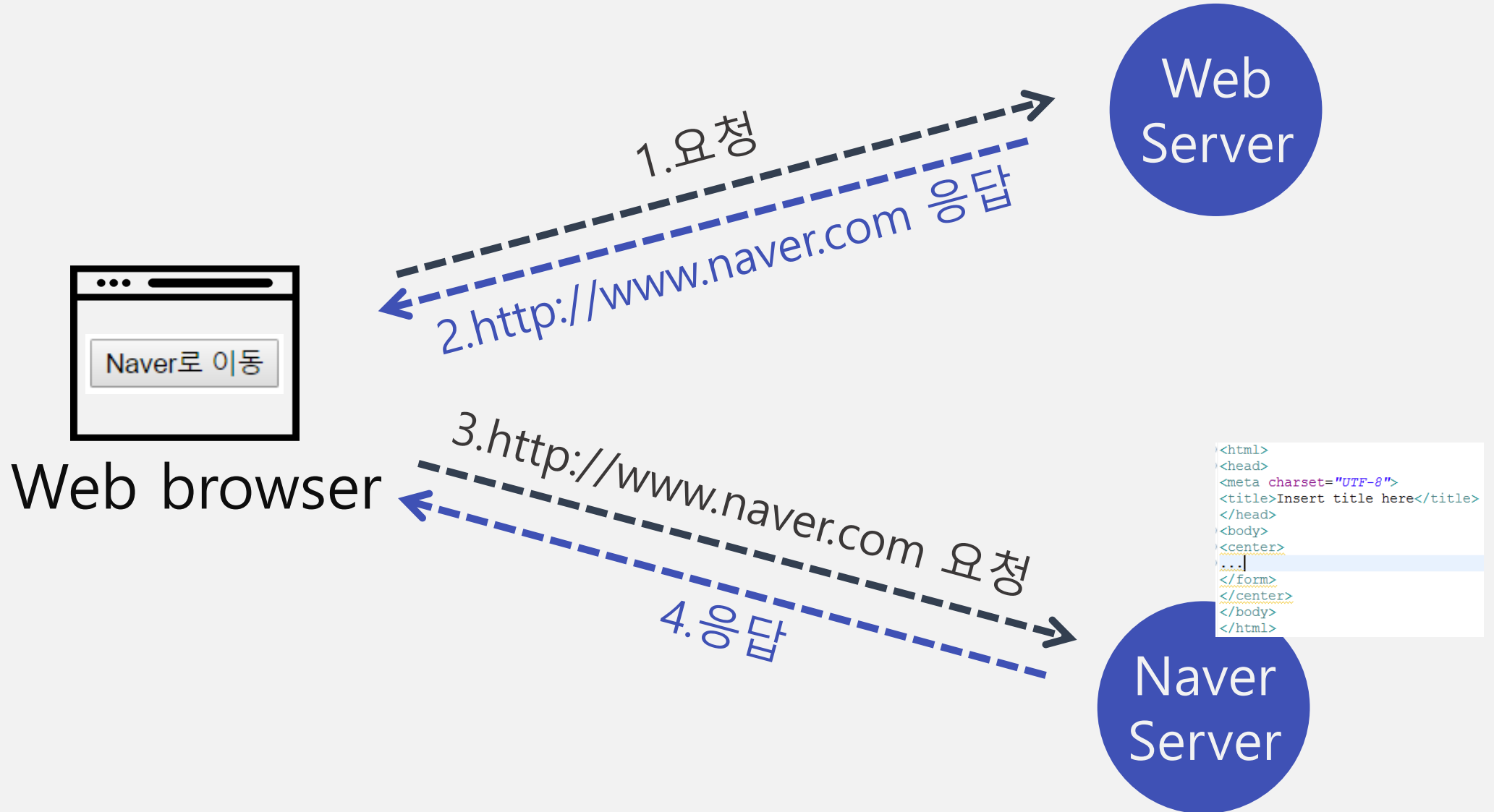
현재 실행 중인 JSP page의 실행을 중단하고 다른 Web page가 대신 호출되도록 만드는 기능



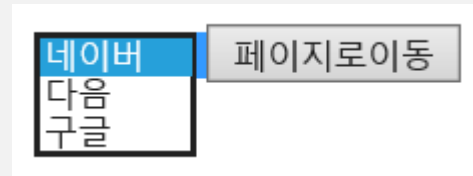
페이지 지
정

sendRedirect() 동작원리

```
<응>  
response.sendRedirect("http://www.naver.com");  
<응>
```



JSP 응용실습



ex12MoveURL.ht
ml



ex12MoveURL.jsp

JSP 응용실습

로그인

<input type="text"/>
<input type="password"/>
<input type="button" value="로그인"/>

ex13Login.html



조건
ID : smart
PASS : 123

ex13Login.jsp



Smart님 환영합니다!

ex13Logintrue.jsp



로그인정보를 다시 확인하세요.
[로그인페이지](#)

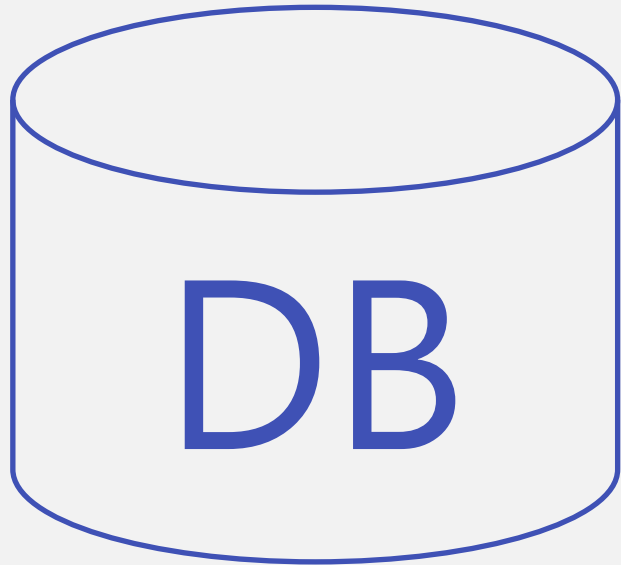
ex13LoginFalse.jsp



로그인

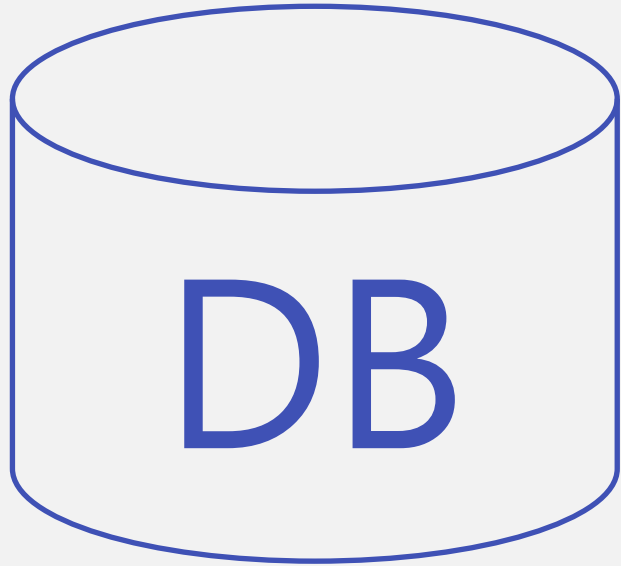
<input type="text"/>
<input type="password"/>
<input type="button" value="로그인"/>

ex13Login.html



데이터베이스에 아래와 같은
조건은 member 라는 테이블
을 만드시오.

Name	Null?	Type
ID		VARCHAR2(50)
PW		VARCHAR2(50)
NICKNAME		VARCHAR2(50)



member 테이블에 아래와 같
은
정보를 입력하시오.

ID	PW	NICKNAME
smart	1234	Jason



ex14Join

1. 오라클 JDBC 라이브러리를 해당 프로젝트 Web Content> WEB-INF > lib 폴더에 복사하시오.
2. 오라클 JDBC 클래스를 로딩하시오.
3. 데이터베이스와 연결하시오.
4. 입력한 ID와 PW를 member테이블에서 검색하시오.
(SQL문 사용)
5. 데이터베이스와 연결끊으시오.

JSP 응용실습

회원가입

ID를 입력해주세요

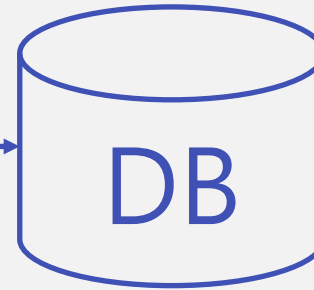
PW를 입력해주세요

NICKNAME을 입력해주세요

회원가입

ex14Join.htm

|



TRUE

FALSE

Smart님!
회원가입을 축하드립니다!

ex14Jointrue.jsp

가입도중 문제가 발생하였습니다. 다시 시도해주세요.

[회원가입페이지](#)

ex14JoinFalse.jsp



1. 오라클 JDBC 클래스를 로딩하시오.
2. 데이터베이스와 연결하시오.
3. 입력한 PW와 NickName을 수정하시오.
(SQL문 사용)
4. 데이터베이스와 연결끊으시오.

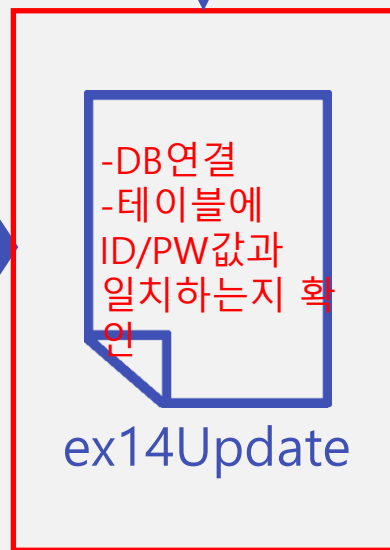
Ex14Update

JSP 응용실습

회원정보수정

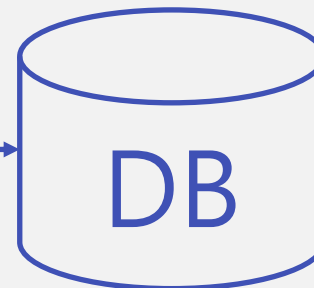
<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="button" value="회원가입"/>

ex14Update.html



TRUE

FALSE



SMSM님
회원정보수정이
완료되었습니다.

ex14UpdateTrue.jsp

정상적으로 이루어지지
않았습니다.
다시 시도해주세요.

[회원정보수정페이지](#)

ex14UpdateFalse.jsp



Ex14delete

1. 오라클 JDBC 클래스를 로딩하시오.
2. 데이터베이스와 연결하시오.
3. 입력한 ID를 통해 PW를 찾고 일치한지 판단하여 삭제하시오.
(SQL문 사용)
4. 데이터베이스와 연결끊으시오.

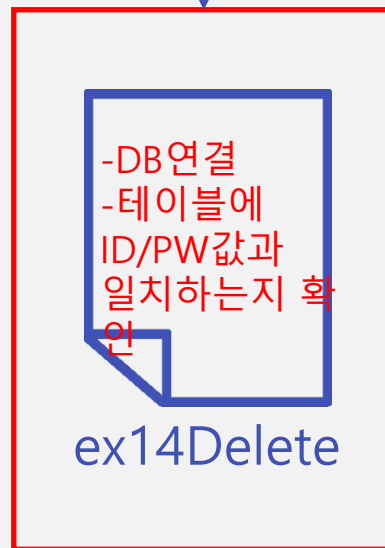
JSP 응용실습

회원탈퇴

ID를 입력해주세요
PW를 입력해주세요
회원탈퇴

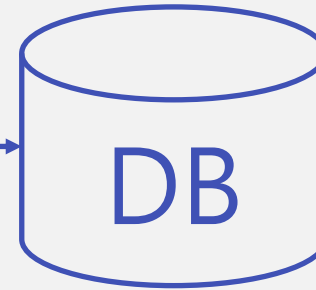
ex14Delete.htm

|



TRUE

FALSE



회원탈퇴가
완료되었습니다.

Ex14DeleteTrue.jsp

비밀번호가 일치하지 않아
탈퇴가 중단되었습니다.

[회원탈퇴페이지](#)

Ex14DeleteFalse.jsp

Thank
you.