



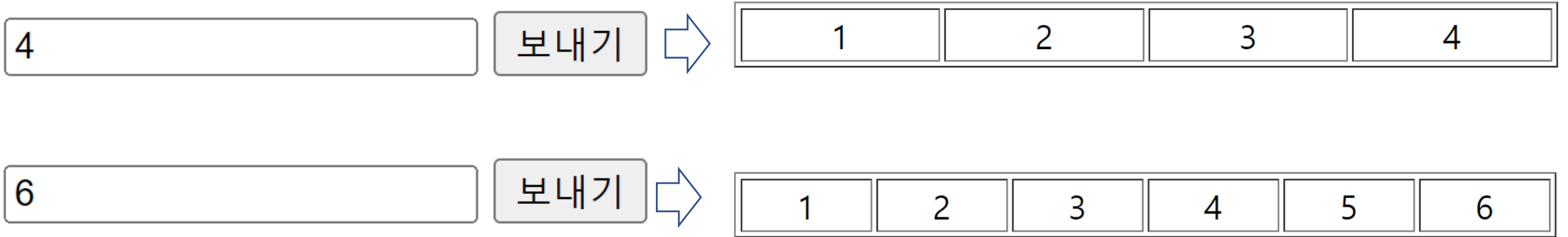
JSP/Servlet

ch3. JSP

1. JSP의 필요성과 특징을 이해한다.
2. JSP 구성요소를 알고 사용할 수 있다.

JSP는 왜 필요할까?

클라이언트 요청에 따라 테이블 셀 생성!



JSP는 왜 필요할까?

```
@WebServlet("/MakeTable")  
public class Ex06MakeTable extends HttpServlet {
```

```
    protected void service(HttpServletRequest request,
```

```
        String data=request.getParameter("room");  
        int room= Integer.parseInt(data);
```

Business Logic

```
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();
```

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

Presentation Logic

```
}
```

JSP는 왜 필요할까?

Servlet의 특징

- ✓ 자바코드로 구현한 후 컴파일하고 배포해야 한다.
- ✓ 코드가 수정되면 다시 컴파일하고 배포해야 한다.
-> 생산성 저하
- ✓ HTML 코드를 Servlet 클래스 내에서 문자열("")로 작성해야 한다.
-> Web Designer도 서블릿 클래스에서 html 코드를 작성해야 함
- ✓ 프로그램이 복잡해지면 출력되는 HTML페이지를 상상하기 어렵다

JSP는 왜 필요할까?

Servlet

```
@WebServlet("/MakeTable")
public class Ex06MakeTable extends HttpServlet {

    protected void service(HttpServletRequest request,

        String data=request.getParameter("room");
        int room= Integer.parseInt(data);

        response.setContentType("text/html;");
        PrintWriter out = response.getWriter();

        out.print("<html>");
        out.print("<body>");
        out.print("<table border='1'>");
        out.print("<tr>");
        for(int i=0;i<room;i++) {
            out.print("<td>"+(i+1)+"</td>");
        }
        out.print("</tr>");
        out.print("</table>");
        out.print("</body>");
        out.print("</html>");

    }
```

Java 코드에 html 작성

JSP

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
    <%
        String data=request.getParameter("room");
        int room= Integer.parseInt(data);
    %>

    <table border='1'>
        <tr>
            <%for(int i=0;i<room;i++) { %>
                <td><%=i %></td>
            <%}%>
        </tr>
    </table>
</body>
</html>
```

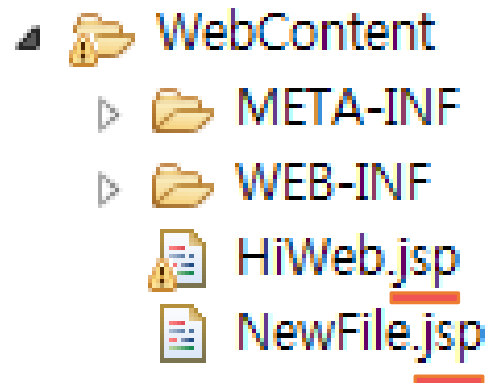
Html에 Java 코드 작성

Java Server Page

HTML내에 Java 코드를 작성하고
웹 서버에서 웹 페이지를 생성하여 웹브라우저에 돌려주는
자바를 기반으로 한 스크립트 언어

JSP란?

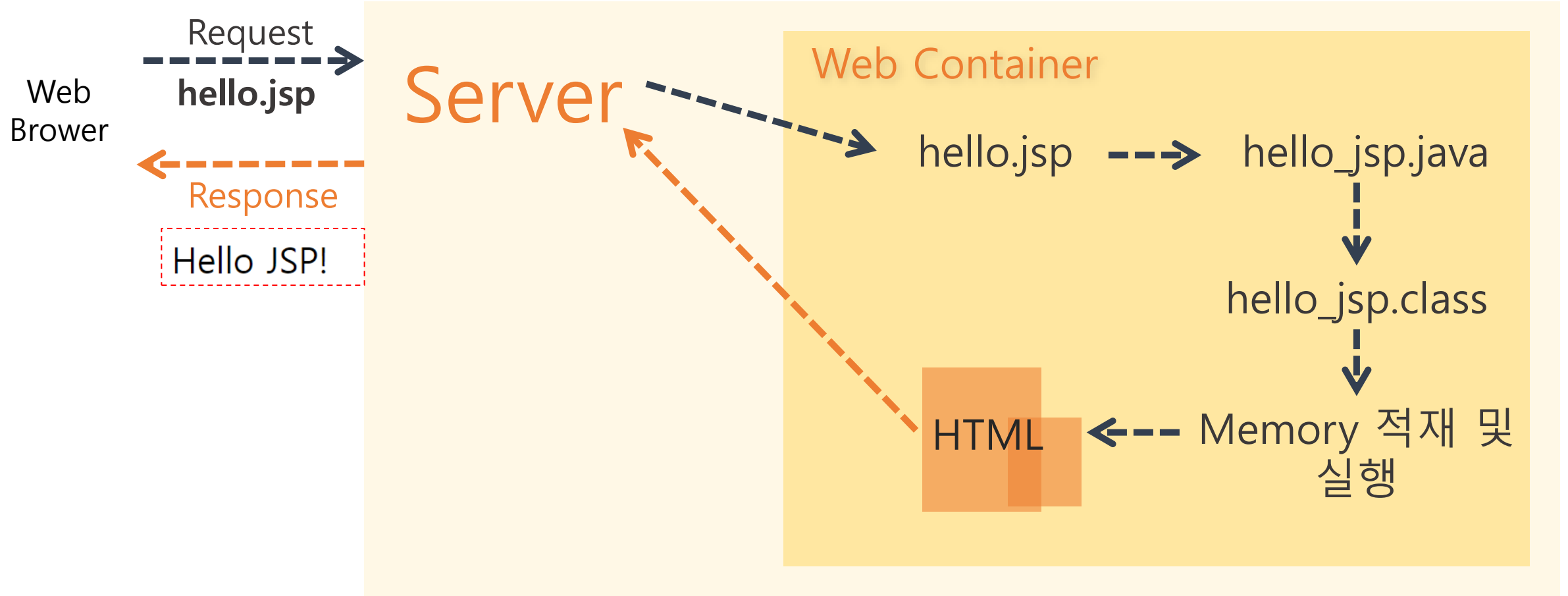
✓ .jsp 확장자를 가짐



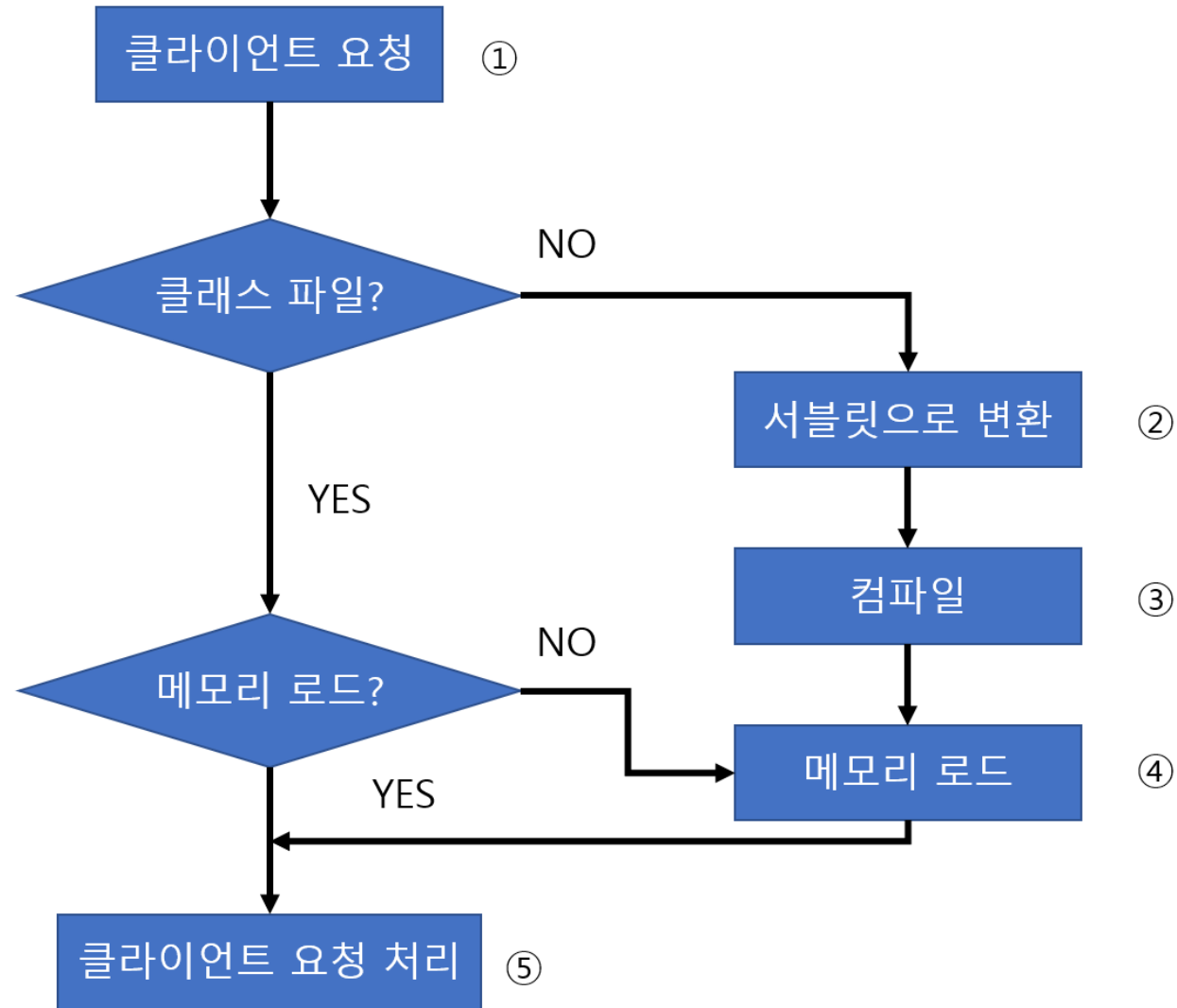
✓ 동적으로 작동하여 응답은 html을 이용함

✓ jsp -> 서블릿(.java) -> 클래스(.class) -> html으로 변환되어 실행됨

JSP Life Cycle



JSP Life Cycle



JSP Life Cycle

* 서블릿으로 변환된 파일 확인 *


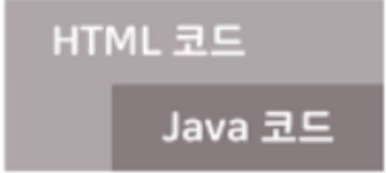
> jsp > .metadata > .plugins > org.eclipse.wst.server.core > tmp0 > work > Catalina > localhost > jsp-study > org > apache > jsp
↓
이클립스 워크스페이스

 hello_jsp.class

 hello_jsp.java

```
out.write("WrWn");
out.write("<!DOCTYPE html>WrWn");
out.write("<html>WrWn");
out.write("<head>WrWn");
out.write("<meta charset=W\"UTF-8W\">WrWn");
out.write("<title>Insert title here</title>WrWn");
out.write("</head>WrWn");
out.write("<body>WrWn");
out.print("Hello JSP!" );
out.write("WrWn");
out.write("</body>WrWn");
out.write("</html>");
```

JSP vs Servlet

	장점	단점	형식
서블릿	복잡한 로직 구현에 적합함	JAVA 코드 중심 화면작성 작업과 수정이 어려움	
JSP	HTML 코드 중심 -> 화면작성 작업, 수정용이	소스보기로 소스가 공개되어 중요정보에 부적합	

로직 구현은 Servlet 클래스에, HTML 중심의 코드는 JSP에

JSP 구성요소

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
14/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)

스크립트릿 (Scriptlet)

`<% %>`

JSP 페이지 내부에 자바(Java) 소스코드를 삽입한다.

표현식 (Expression)

`<%= %>`

출력결과(html 요소내)에 값을 넣기 위해 사용한다.

지시자 (Directive)

<%@ %>

JSP에 대한 설정 정보 또는 JSP 페이지에 다른 문서를 포함시킬 때 사용한다.

선언문 (Declaration)

<%! %>

JSP 페이지 내부에 자바 메소드, 변수를 선언할 때 사용한다.

Scriptlet(스크립트릿)

1. 스크립트릿 <% %>

- JSP문서 안에 JAVA코드를 넣기 위해 사용함.

```
<body>
```

```
<%
```

```
    int num1 = 20;
```

```
    int num2 = 10;
```

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

```
    out.println(num1 + "+" + num2 + "=" + sum);
```

```
%>
```

```
</body>
```

실행결과

20+10=30

Expression(표현식)

2. 표현식 <%= %>

- Web Browser에 결과 값을 출력하기 위해 사용함.

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

```
%>
```

```
<b>두수의 합:</b>
```

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

```
<b>두수의 곱:</b>
```

```
<%=num1*num2 %><br>
```

```
<b>두수의 제곱근</b>
```

```
<%=Math.sqrt(num1+num2) %>
```

1) 상수나 변수

2) 연산자

3) 메소드호출

실행결과

두수의 합: 30

두수의 곱: 200

두수의 제곱근 5.477225575051661

실습 1. 1~100까지의 합을 구하여 화면에 출력하시오.

1~100까지의 합 : 5050

Ex01hap100.jsp

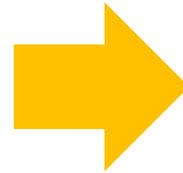
실습2. 방이 10개인 table을 화면에 출력하시오(*for문 사용하기)

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

Ex02Table.jsp

실습3. html에서 정수 2개와 연산기호를 입력받아
jsp에서 연산결과를 출력하시오.

Ex03Input.html



$$5 + 2 = 7$$

Ex03Return.jsp

Declaration(선언문)

3. 선언문 <%! %>

- 전역 변수나 메소드를 선언하여 사용함.

```
<%!  
int global_cnt = 0; | 전역변수로 선언  
%>
```

최초실행

local_cnt: 1
global_cnt: 1

```
<%  
int local_cnt = 0; | 지역변수로 선언
```

```
out.print("<br> local_cnt: ");  
out.print(++local_cnt);
```

```
out.print("<br> global_cnt: ");  
out.print(++global_cnt);
```

```
%>
```

새로고침실행

local_cnt: 1
global_cnt: 2

Declaration(선언문)

```
<%!  
    int global_cnt = 0;  
%>  
  
<%  
    int local_cnt = 0;  
  
    out.print("<br> local_cnt: ");  
    out.print(++local_cnt);  
  
    out.print("<br> global_cnt: ");  
    out.print(++global_cnt);  
%>
```

```
public final class declare_jsp extends org.apache.jasper.runtime.HttpJspBase  
    implements org.apache.jasper.runtime.JspSourceDependent,  
                org.apache.jasper.runtime.JspSourceImports {  
  
    int global_cnt = 0;  
  
    :  
  
    public void _jspService(final javax.servlet.http.HttpServletRequest request,  
final javax.servlet.http.HttpServletResponse response)  
        throws java.io.IOException, javax.servlet.ServletException {  
  
        :  
  
        int local_cnt = 0;  
  
        out.print("<br> local_cnt: ");  
        out.print(++local_cnt);  
  
        out.print("<br> global_cnt: ");  
        out.print(++global_cnt);  
  
        :  
  
    }  
}
```

실습4. 절대값을 return하는 메소드를 다음과 같이 선언문으로 작성.
해당 메소드를 호출하여 아래와 같이 화면에 출력하시오.

```
<%!  
    int abs(int num){  
        if(num>0){  
            return num;  
        }else {  
            return -num;  
        }  
    }  
%>
```

5의 절대값 : 5
-5의 절대값 : 5

실행 결과

Ex04abs.jsp

실습5. num1,num2,sum을 아래와 같이 선언하고
정수가 홀수인지 짝수인지 판별하는 메소드를 작성하시오.
그 후 sum값의 홀짝여부를 화면에 출력하시오.

```
<%  
    int num1 = 10;  
    int num2 = 21;  
    int sum = num1+num2;  
<%!  
    public boolean isOdd(int num){  
  
    }  
%>
```



sum은 홀수입니다!

Ex05isOdd.jsp

4. 지시자 <%@ %>

- Web Container가 JSP를 Servlet class로 변환할 때 필요한 정보를 기술하기 위해 사용함.

page
지시자

taglib
지시자

include
지시자

Directive(지시자: <%@ %>)

4-1 . page지시자

- JSP페이지의 전체적인 환경설정을 할 때 사용함.

```
<%@ 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 속성을 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 사용

Directive(지시자: <%@ page %>)

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

Servlet 코드로 변환

```
try {
```

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

Directive(지시자: <%@ page %>)

<%@page import="java.util.Calendar"%>



```
<%  
    Calendar cal = Calendar.getInstance();  
    int year = cal.get(Calendar.YEAR);  
    int month = cal.get(Calendar.MONTH)+1;  
    int day = cal.get(Calendar.DATE);  
  
    out.println(year + "년 " + month + "월 " + day + "일");  
%>
```

Directive(지시자: <%@ page %>)

실습: devidebyZero.jsp

0과 2를 나눠볼까요?

<%=2/0 %>

실행결과

0으로 나누는 것은 불가능

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.serviceJspFile(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.serviceJspFile(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

Directive(지시자: <%@ page %>)

실습. 0으로 나눌 때 오류페이지 출력하도록 코드 수정

1) dividebyZero.jsp 파일에 다음 지시어 추가

```
<%@page errorPage = "errorPage.jsp" %>
```

2) errorPage.jsp 에 다음과 같은 지시어와 코드 추가

```
<%@page isErrorPage="true" %>
```

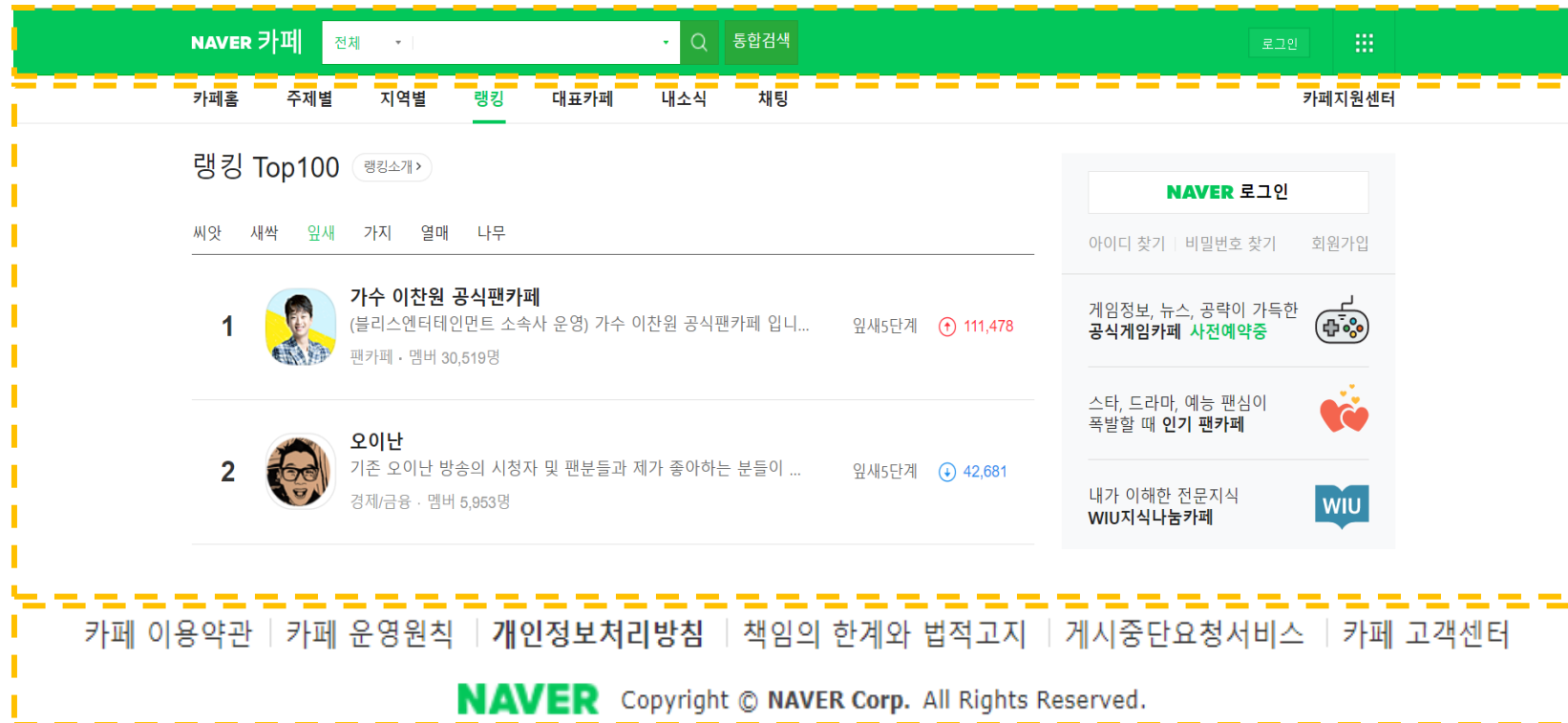
다음과 같은 예러가 발생하였습니다.

```
<%= exception.getMessage() %>
```

Directive(지시자: <@ include file="..">)

4-2 . include지시자

- 현재 페이지에 다른 파일의 내용을 삽입할 때 사용함.
- 페이지끼리 변수를 공유해서 사용하고자 할 때



nav

현재페이지

footer

Directive(지시자: <@ include file="..">)

<%@include file="today.jsp" %> 실습.

두번째 페이지로

Today: 2022년 8월 10일

today.jsp

첫번째 페이지로

Today: 2022년 8월 10일

4-3 . taglib지시자

- 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>
```

```
<%  
    for (int i = 1; i <= 10; i ++ ) {  
%>  
    <%= i %><br/>  
  
<%  
    }  
%>
```



```
<c:forEach var="i" begin="1" end="10" step="1">  
    <c:out value="${i}"/>  
    <br/>  
</c:forEach>
```

5. 주석 <%-- --%>

- 프로그램 설명, 테스트 목적으로 사용되는 태그

```
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>
```

```
<%-- Java 주석 --%>
```

```
</body>
```

```
</html>
```