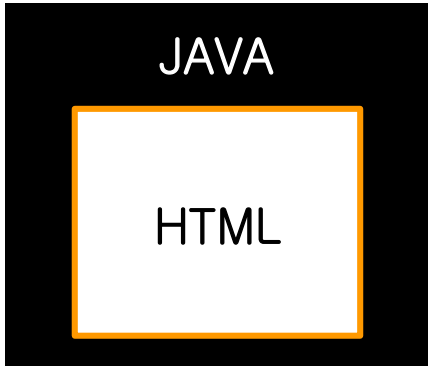



# **JSP**

## **(Java Server Page)**

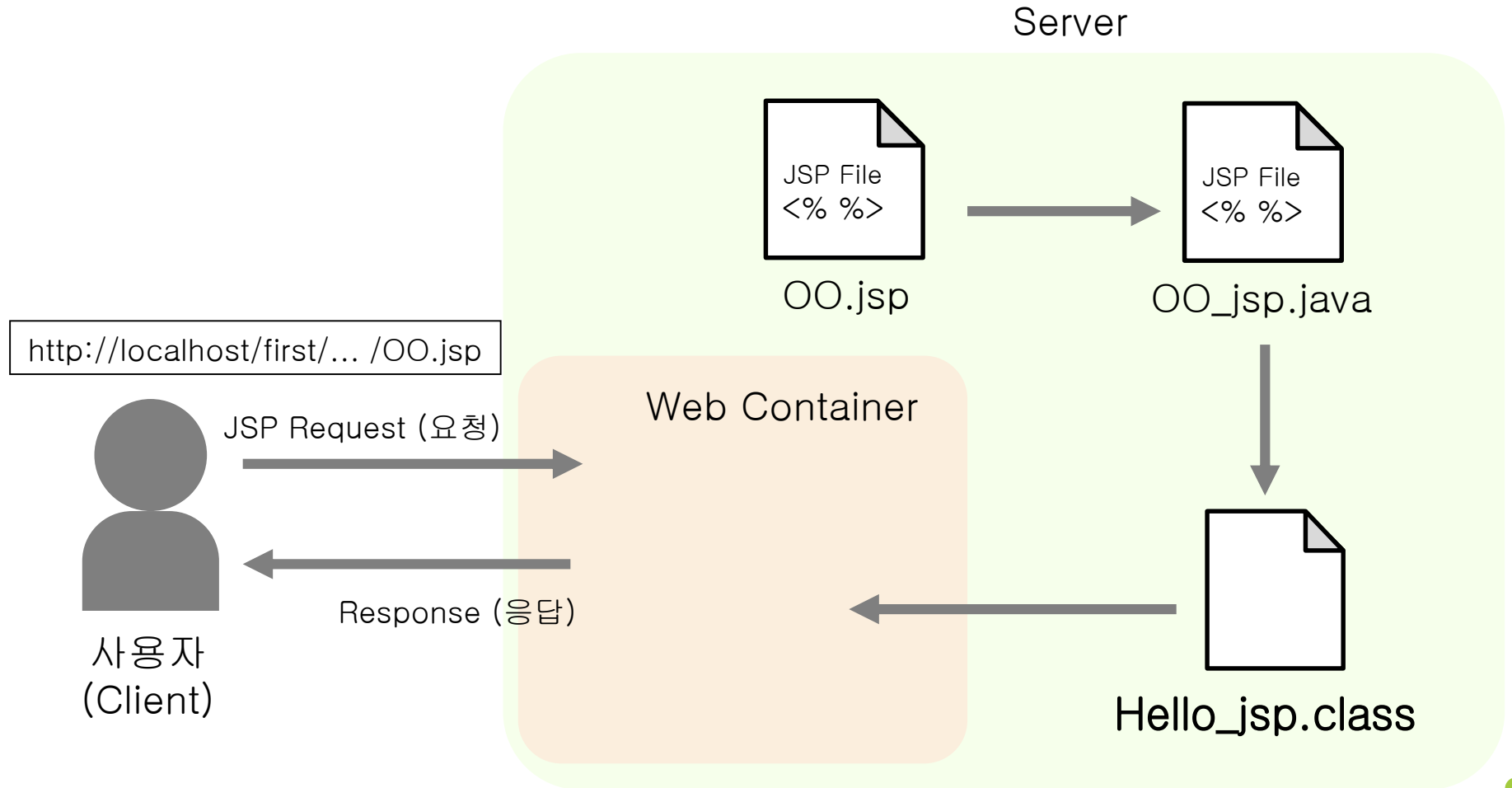
## Servlet과 JSP 비교

	Servlet	JSP
형태	Java 코드에 HTML 코드를 삽입 	HTML 코드에 Java 코드를 삽입 
예시	<code>out.println( "&lt;HTML&gt;" );</code>	<code>&lt;% for (int i=0; i&lt;10; i++) { %&gt;</code>
특징	Business 로직 처리에 적합	화면 로직 처리에 적합

※ JSP 기술의 목표는 Servlet의 Business 로직으로부터  
화면 로직을 분리하는데 있다

# JSP (Java Server Page)

## JSP 실행 방식



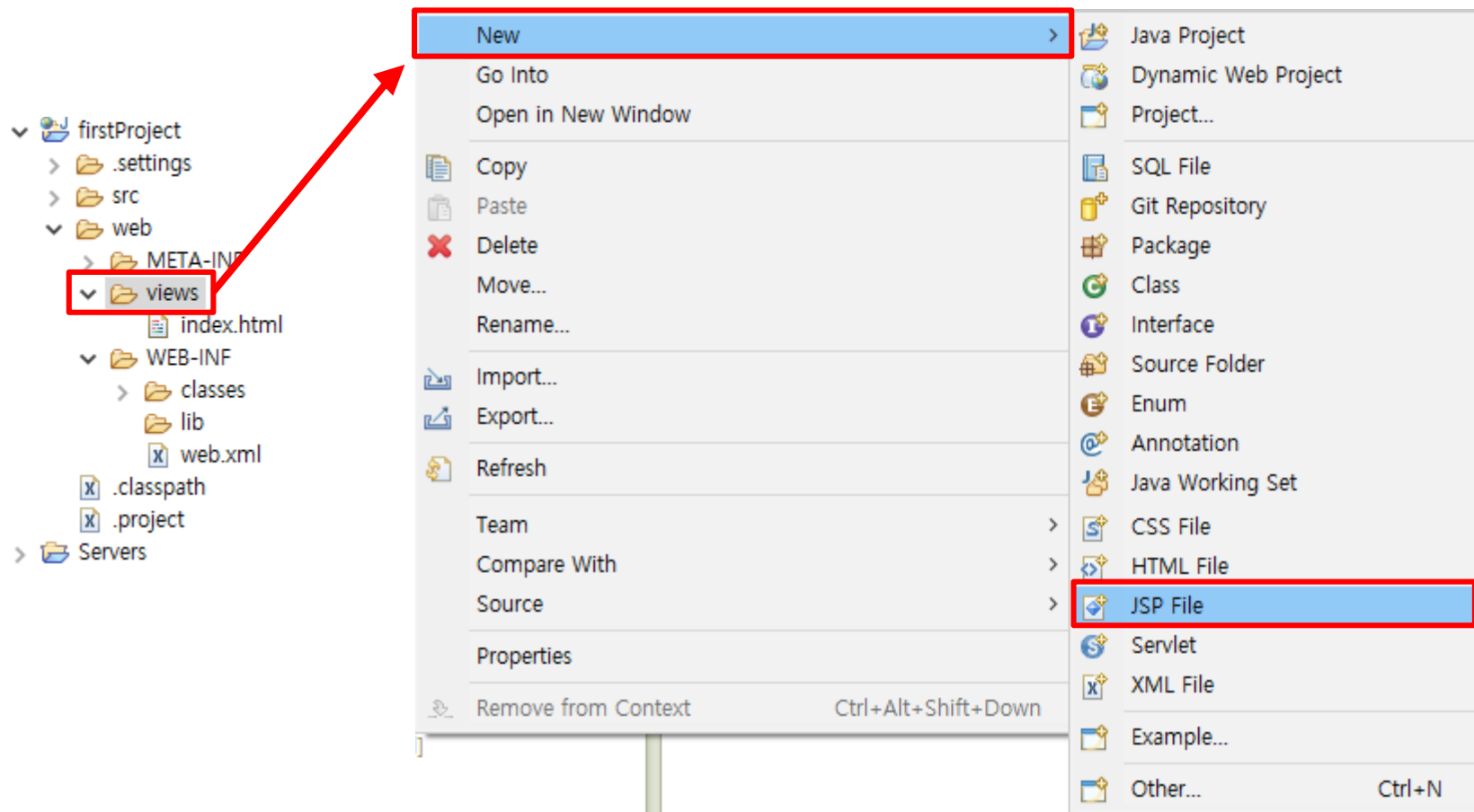
## JSP 특징

- I. JSP 파일이 변경되지 않는다면, '.jsp' 파일에 대한 컴파일은 다시 일어나지 않는다.
- II. JSP 파일이 변경될 때 마다, Web Container는 translation, compile, load, initialization 과정을 수행한다.  
  
※ 주의 : 구 버전의 JSP 파일을 overwrite 할 경우 제대로 반영이 되지 않는 경우가 발생할 수 있다.
- III. JSP 파일의 배포 환경(위치)은 HTML과 동일하다  
= WEB\_ROOT 폴더 하단.

# JSP (Java Server Page)

## JSP 만들기

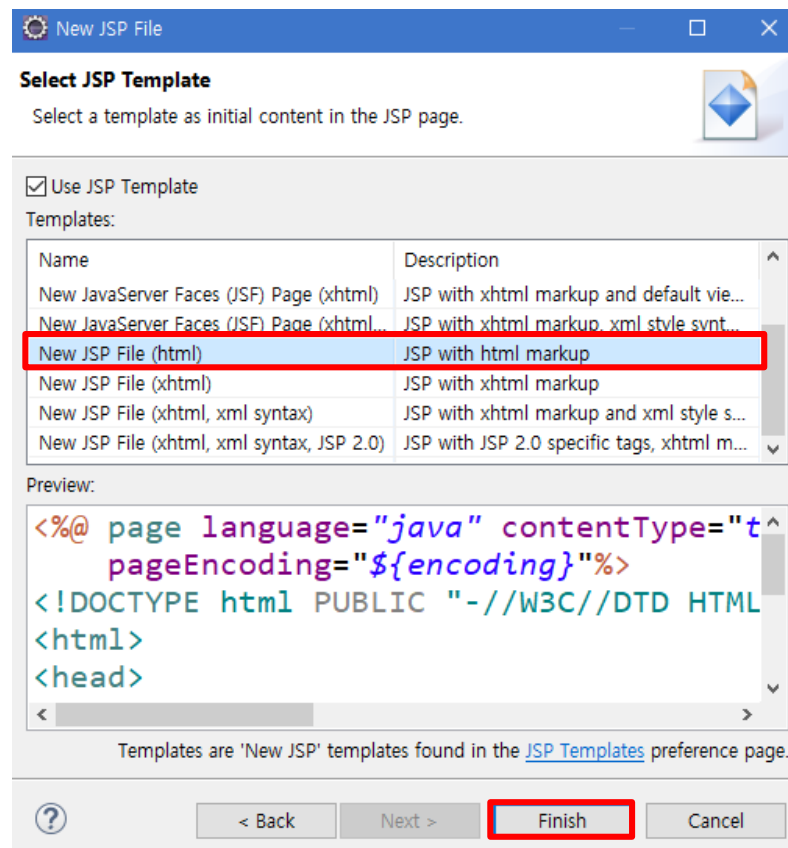
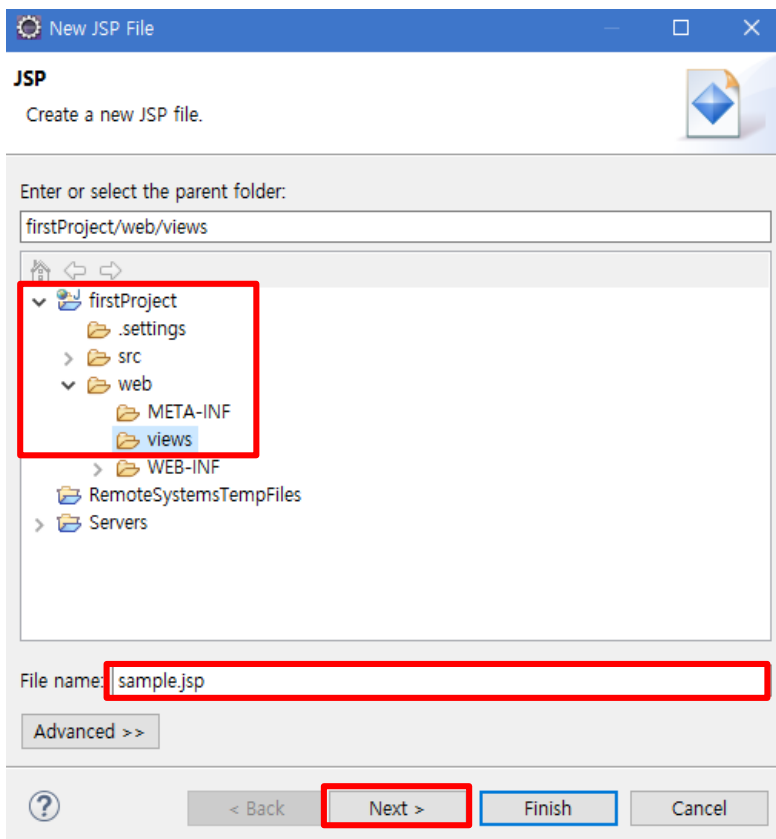
웹 프로젝트 내 web/views 를 선택 후, 우클릭하여 JSP 파일 생성



# JSP (Java Server Page)

## JSP 만들기

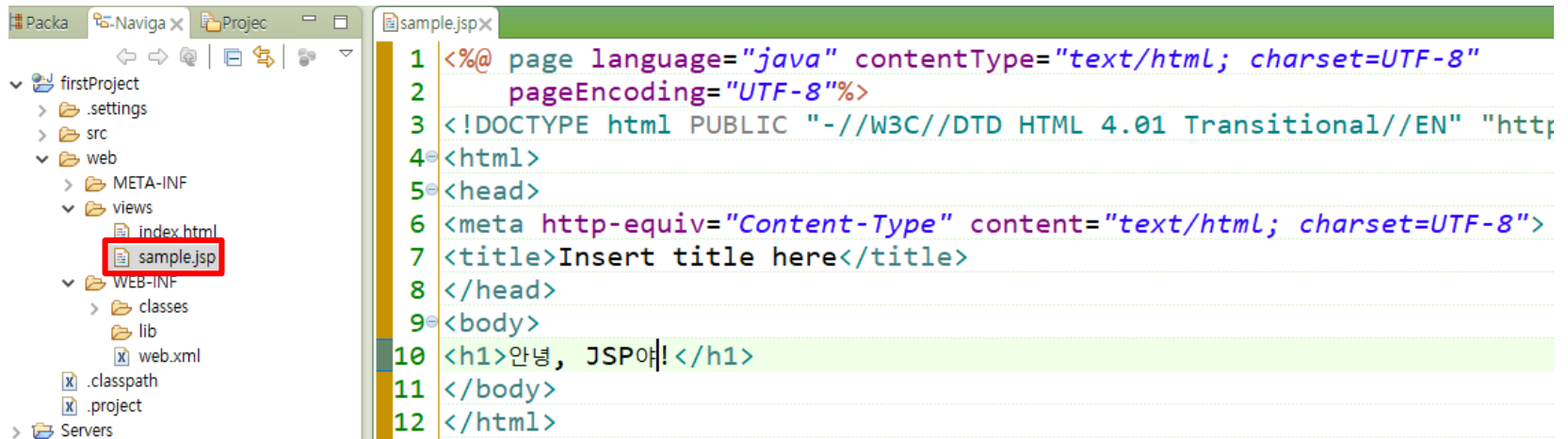
생성될 경로를 확인하고 Next, New JSP File 양식 선택 후 Finish




# JSP (Java Server Page)

## JSP 만들기

생성된 JSP 파일을 확인하고 서버를 실행하여 접근 확인



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>안녕, JSP야!</h1>
11 </body>
12 </html>
```



localhost:8800/first/views/sample.jsp

안녕, JSP야!

# JSP Elements



## JSP Element 표기법

Comments tag	<code>&lt;%-- 주석 --%&gt;</code>
Directive tag	<code>&lt;%@ 지시자 %&gt;</code>
Declaration tag	<code>&lt;%! 선언문 %&gt;</code>
Scriptlet tag	<code>&lt;% 코드 %&gt;</code>
Expression tag	<code>&lt;%= 표현식 %&gt;</code>

## JSP Element 표기법 예제

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

<%!
    // JSP 선언 태그입니다.
    public static final String NAME = "JSP World";
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Elements 실습</title>
</head>
<body>
<!-- JSP 주석 태그 입니다. --%>

<%
    // JSP 스크립트릿 태그입니다.
    out.println("Welcome :");
    String name = request.getParameter("name");
    if(name == null || name.isEmpty()){
        name = NAME;
    }
%>

<!-- JSP 값 표현 태그입니다. -->
<h3>Hello, <%= name %></h3>
</body>
</html>
```

JSP 지시자 태그

JSP 선언 태그

JSP 주석 태그

JSP 스크립트릿 태그

JSP 값 표현 태그

JSP Elements 예제 = web/views/jspPractice/jspElements.jsp

## Comments tag 종류에 따른 컴파일 여부

- HTML 주석

내부에서 `out.write();` 로 변환되나 화면에는 보이지 않는다.

```
<!-- HTML 주석입니다 -->  
→ out.write(" <!-- HTML 주석입니다 --> \r\n");
```

- JSP 주석 태그

JSP 파일 내에만 존재하고, Servlet 코드에는 포함되지 않는다.

```
<%-- JSP 주석 태그 입니다. --%>
```

- Java 주석 태그

변환된 Servlet 코드에는 포함되지만 HTTP 응답으로 전송되지 않는다.

```
<%-- //Java 주석입니다. --%>  
→ //Java 주석입니다.
```

## Directive tag

- JSP page 전체에 영향을 미치는 정보를 기술할 때 쓰인다.

```
<%@ 지시자 [속성명="value"] ... %>
```

- 지시자 종류

page, include, taglib 3종류로 나뉜다.

```
<%@ page import="java.io.*"%>  
<%@ include file="header.html"%>  
<%@ taglib uri="/WEB-INF/tags/abc" prefix="abc" %>
```

## Declaration tag

- Servlet 클래스의 멤버변수/메소드에 해당하는 코드를 작성할 때 사용된다.

멤버 변수 선언

```
<%! public static final String DEFAULT_NAME="홍길동"; %>  
<%! int counter = 0; %>
```

멤버 메소드 선언

```
<%!  
    public String getName(HttpServletRequest request) {  
        return request.getParameter("name");  
    }  
%>
```

## Scriptlet tag

- `_jspService` 메소드의 로컬 변수와 코드를 작성할 때 사용된다.

로컬 변수 선언

```
<% int i = 0; %>
```

자바 코드 내용 기술

```
<% if ( i > 10 ) { %>
```

i가 10보다 큼니다.

```
<% } else { %>
```

i가 10보다 작습니다.

```
<% } %>
```

## Expression tag

- Servlet 코드에서 out.print() 의 역할을 수행한다.

예제

현재 시간은 <%= new java.util.Date() %> 입니다.

※ 표현 태그에서는 ‘;’ 을 붙이지 않는다

<%= new java.util.Date(); %>

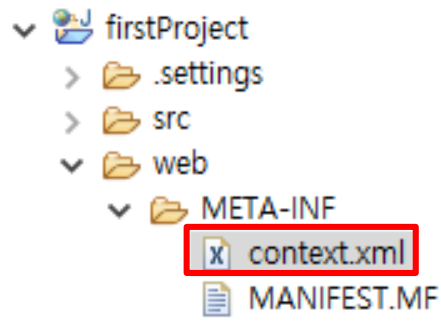
Servlet  
변환

out.print( new java.util.Date(); ); → Syntax Error !!

## JSP도 Servlet이다!

JSP도 컴파일 시 Servlet으로 변환되어 서비스 된다.

해당 과정을 직접 보기 위해 먼저 web/META-INF/ 경로에 context.xml 파일을 생성하여 다음과 같이 기록한다.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more contrib
    ownership. The ASF licenses this file to You under the Apache License, Ver
    License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by a
    WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the L
    --><!-- The contents of this file will be loaded for each web application -->
<Context workDir="D:\workspace\firstProject\web\WEB-INF\jspwork">

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->
    <Resource auth="Container" driverClassName="oracle.jdbc.OracleDriver" maxActiv
        url="jdbc:oracle:thin:@127.0.0.1:1521:xe" username="student" />
</Context>
```



## JSP도 Servlet이다!

context.xml은 웹 애플리케이션 서버에서 사용할 자원을 설정하는 파일이다. 이 중 Context workDir 속성은 컴파일 된 JSP class 파일이 위치할 경로를 가리킨다. 해당 경로는 자신의 프로젝트 내 WEB-INF 경로에 맞게 작성하자

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more contrib
    ownership. The ASF licenses this file to You under the Apache License, Ver
    License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by a
    WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the L
    --><!-- The contents of this file will be loaded for each web application -->
<Context workDir="D:\workspace\firstProject\web\WEB-INF\jspwork">

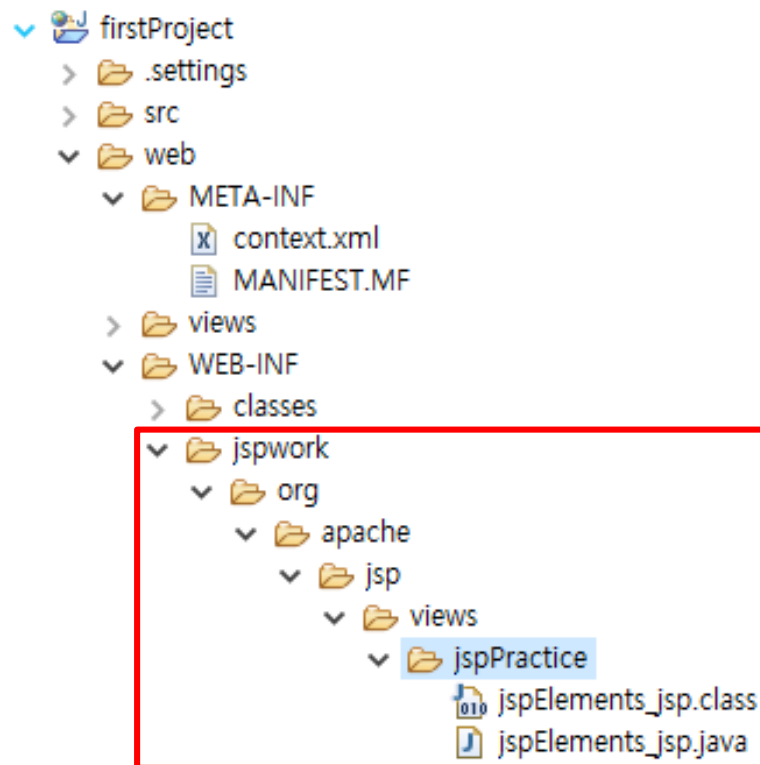
    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->
    <Resource auth="Container" driverClassName="oracle.jdbc.OracleDriver" maxActiv
        url="jdbc:oracle:thin:@127.0.0.1:1521:xe" username="student" />
</Context>
```

## JSP도 Servlet이다!

설정이 끝났다면 서버를 재실행하고 Project Explorer를 갱신하여 생성된 소스코드를 확인해보자.



## JSP Elements - Java 코드 변환 내용

jspElements.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<!--
    // JSP 선언 태그입니다.
    public static final String NAME = "JSP World";
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Elements 실습</title>
</head>
<body>
<!-- JSP 주석 태그입니다. -->
<%
    // JSP 스크립트릿 태그입니다.
    out.println("Welcome :");
    String name = request.getParameter("name");
    if(name == null || name.isEmpty()){
        name = NAME;
    }
%>
<!-- JSP 값 표현 태그입니다. -->
<h3>Hello, <%= name %></h3>
</body>
</html>
```

jspElements\_jsp.java

```
response.setContentType("text/html; charset=UTF-8");
pageContext = _jspxFactory.getPageContext(this, request, response,
    null, true, 8192, true);
_jsp_page_context = pageContext;
```

```
public static final String NAME = "JSP World";
```

```
out.println("Welcome :");
String name = request.getParameter("name");
if(name == null || name.isEmpty()){
    name = NAME;
}
```

```
out.write("<h3>Hello, ");
out.print( name );
out.write("</h3>\r\n");
```

※ JSP 주석 태그는 컴파일 시 포함되지 않는다.

# JSP 내장 객체

## JSP 내장 객체란?

JSP에서 기본적으로 제공하는 객체들로 request, response, out 등 Scriptlet tag와 Expression tag에서 사용할 수 있도록 암시적으로 선언된 객체를 뜻한다.

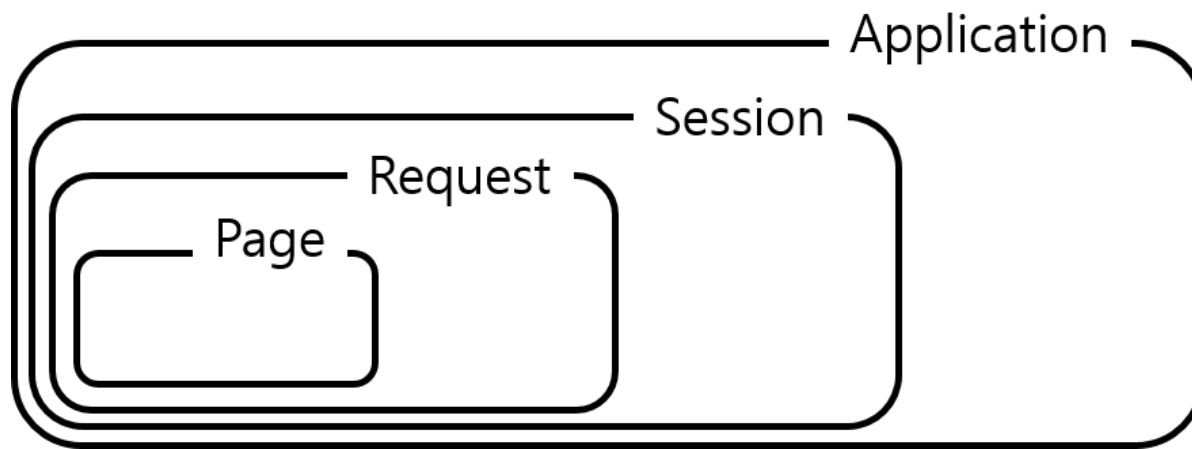
```
----- jspElements.jsp -----
<%
    // JSP 스크립트릿 태그입니다.
    out.println("Welcome :");
    String name = request.getParameter("name");
    if(name == null || name.isEmpty()){
        name = NAME;
    }
%>
```

## JSP 내장 객체의 종류

내장 객체 명	설명
request	HttpServletRequest 객체 참조 변수
response	HttpServletResponse 객체 참조 변수
out	JspWriter 객체 참조 변수
session	HttpSession 객체 참조 변수
application	ServletContext 객체 참조 변수
page	현재 JSP 페이지에 대한 참조 변수
exception	발생 하는 Throwable 객체에 대한 참조 변수

## JSP 내장 객체의 영역

영역	설명
page	하나의 JSP페이지를 처리할 때 사용되는 영역
request	하나의 요청을 처리할 때 사용되는 영역
session	하나의 브라우저와 관련된 영역
application	하나의 웹 어플리케이션과 관련된 영역



## Request 주요 메소드

메소드 명	설명
getParameter(name)	name 파라미터의 값을 리턴한다
getParameterValues(name)	name 파라미터의 값을 배열 형태로 리턴한다. (checkbox 등에 쓰임)
getParameterNames()	요청에 포함된 파라미터 이름들을 리턴한다.
getMethod()	현재 요청 방식을 리턴한다. (GET, POST)
getSession()	현재 세션 객체를 리턴한다.
setCharacterEncoding()	클라이언트에서 서버로 전달된 값을 지정한 문자셋으로 변경한다.



## Response 주요 메소드

메소드 명	설명
sendRedirect(url)	응답 결과를 요청으로 하여 지정된 url에 재전송한다.
setStatus(int status_code)	응답으로 전송될 상태 코드를 설정한다. 성공일 경우 기본값은 '200', OK이다.
sendError(int status_code)	에러가 발생할 경우 응답 헤더에 상태 코드를 설정한다.
setContentType(String)	서버에서 클라이언트로 전달될 값의 데이터 타입을 설정한다.

## HTTP Request 전송의 GET 방식과 POST 방식

HTTP 프로토콜을 통해 데이터를 전송할 때 보통 두 가지의 Request Method를 사용하는데 바로 GET 방식과 POST 방식이다.

GET 방식은 요청한 정보와 함께 전달되는 파라미터 값이 URL 내부에 쿼리 스트링(query string)으로 저장되어 보내진다.

POST 방식은 서버로 파라미터 값이 전달될 때, HTTP 메시지 바디 안에 query string이 저장되어 보내진다.

## Query String이란?

사용자가 서버로 데이터를 전달할 때 전송된 데이터들을 URL의 뒷부분에 '?'로 구분 지어 전송하는 것을 말한다.

key 는 input 태그의 name 값을, value는 input 태그의 value 값을 뜻한다.  
'?' 는 Query String의 시작을 의미하며, '&' 는 각 데이터 간의 구분 자를 뜻한다.

### [표현 식]

`http://localhost:8800/first/test.jsp?key=value`

### [표현 예제]

`http://localhost:8800/first/test.jsp?id=sample&pwd=sample`

## GET 방식 예제

The screenshot shows a web browser window with the address bar displaying `localhost:8800/first/views/jspPractice/getMethod.html`. The page content is titled "GET 방식 전송 실습" and contains a form with two input fields: "ID : test" and "PASSWORD : ....", followed by a "제출" (Submit) button.

On the right side, the browser's developer tools are open to the "Network" tab. It shows a single request named "getMethod..." with a status of 300 and a duration of 20 ms. The "Waterfall" view shows the request starting at 0 ms and ending at 20 ms. The "Name" column shows "getMethod...", and the "S..." column shows "3...".

GET 송신 예제 = `web/views/jspPractice/getMethod.html`

## GET 방식 예제

Chrome 브라우저에서 F12 클릭!

← → ↻ ⓘ localhost:8800/first/views/jspPractice/GET\_test.jsp?userId=test&userPwd=test ☆ ⋮

GET HTTP Request Headers 정보

Request Method	GET
Protocol	HTTP/1.1
QueryString	userId=test&userPwd=test
Server name	localhost
Server port	8800
Locale	ko_KR
브라우저 정보	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

전달받은 인자 :

userId = test  
userPwd = test

Elements Console Sources Network >> ⋮ ×

Name × Headers Preview Response Cookies Timing

GET... Server: Apache-Coyote/1.1

▼ Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, br  
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7  
Cache-Control: max-age=0  
Connection: keep-alive  
Cookie: JSESSIONID=AE886DE8618D8C06A57A6ACF6B85E4  
Host: localhost:8800  
Referer: http://localhost:8800/first/views/jspPractice/getMethod.html  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

▼ Query String Parameters view source view URL en

userId: test  
userPwd: test

1 request...

GET 수신 예제 = web/views/jspPractice/GET\_test.jsp

## POST 방식 예제

The screenshot shows a web browser window with the address bar displaying `localhost:8800/first/views/jspPractice/postMethod.html`. The page content is titled "Post 방식 전송 실습" and contains a form with two input fields: "ID : test" and "PASSWORD : ....", followed by a "제출" (Submit) button. To the right of the browser window, the Network tab is open, showing a list of network requests. The first request, named "postMet...", is highlighted. The waterfall chart below the list shows the request's timeline, with a total duration of 100.00 ms. The request is categorized as "XHR" (XMLHttpRequest) and "JS" (JavaScript).

POST 송신 예제 = `web/views/jspPractice/postMethod.html`

## POST 방식 예제

localhost:8800/first/views/jspPractice/POST\_test.jsp

### POST HTTP Request Headers 정보

Request Method	POST
Protocol	HTTP/1.1
QueryString	null
Server name	localhost
Server port	8800
Locale	ko_KR
브라우저 정보	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

전달받은 인자 :

userId : test  
userPwd : test

Network Headers

Request Headers

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, br  
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7  
Cache-Control: max-age=0  
Connection: keep-alive  
Content-Length: 24  
Content-Type: application/x-www-form-urlencoded  
Cookie: JSESSIONID=AE8B6DE8618D8C06A57A6ACF6B85EA11  
Host: localhost:8800  
Origin: http://localhost:8800  
Referer: http://localhost:8800/first/views/jspPractice/postMethod.html  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

Form Data

userId: test  
userPwd: test

POST 수신 예제 = web/views/jspPractice/POST\_test.jsp

# JSP 지시자 태그



## page 지시자 태그 사용법

- 여러 개의 page 구문을 사용할 수 있지만, import 속성을 제외하고는 한 페이지에 한 번씩만 선언할 수 있다.
- page 지시어는 JSP 파일의 어느 위치에 와도 상관 없으나, 가장 첫 부분에 사용하는 것이 좋다.

```
<%@ page import="java.io.*"%>  
<%@ page contentType="text/html" %>
```

## import

- 변환될 서블릿 클래스에 필요한 자바 클래스의 import 문을 정의한다.
- java.lang, javax.servlet, javax.servlet.http, javax.servlet.jsp 는 기본적으로 import 되어 있다.
- 여러 package import시 ‘,’ 기호를 이용하여 구분한다.

```
<%@ page import="java.io.*"%>  
<%@ page contentType="text/html" %>
```

## contentType

- MIME 타입과 문자 인코딩을 설정한다.

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

## isErrorPage

- 현재 페이지가 JSP 오류 처리용 페이지인지를 정의한다.
- 값은 true 또는 false(default)이다.
- true인 경우, exception 내장 객체를 사용할 수 있다.

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

## errorPage

- 해당 JSP 페이지가 발생시키는 모든 runtime exception을 처리할 다른 JSP페이지를 지정한다.
- 값은 상대적인 URL이다.

```
<%@ page errorPage="/error/errorForm.jsp" %>
```

## include 지시자 태그

- include 지시자 태그를 사용하면 다른 페이지(JSP, HTML)를 포함할 수 있다.

- 문법

```
<%@ include file="페이지 경로" %>
```

- 사용 예제

```
<%@ include file="footer.html" %>
```

## include 지시자 태그 실습

- 포함될 페이지 – web/views/footer.html

```
<footer> Copyright KH Corp. All rights reserved. </footer>
```

- 포함하는 페이지 – web/views/index.jsp

```
<%@ page language="java" contentType="text/html;  
charset=UTF-8" pageEncoding="UTF-8"%>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>JavaServer Page 실습</title>  
</head>  
<body>  
<h1>Index Page 입니다.</h1>  
<hr>  
<%@ include file="footer.html" %>  
</body>  
</html>
```

## include 지시자 태그 실습

- 실습 테스트



Index Page 입니다.

index.jsp

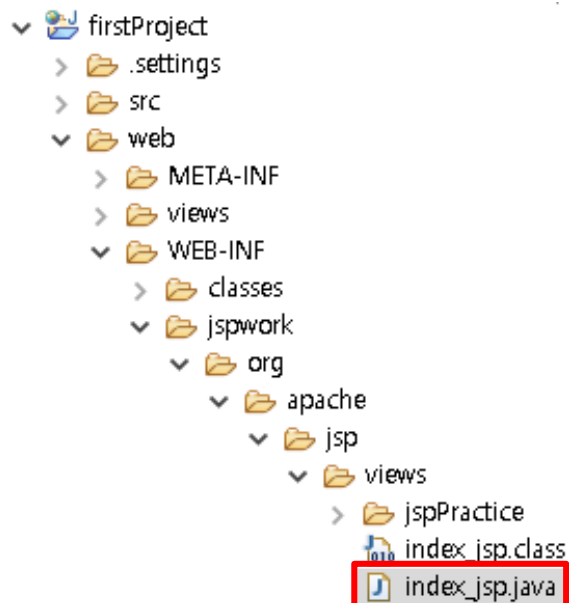
Copyright KH Corp. All rights reserved.

footer.html

예제 = web/views/index.jsp

## include 지시자 태그 실습

- 실습 테스트



index\_jsp.java

```
out.write("<!DOCTYPE html>\r\n");
out.write("<html>\r\n");
out.write("<head>\r\n");
out.write("<meta charset=\"UTF-8\">\r\n");
out.write("<title>Java Server Page 실습</title>\r\n");
out.write("</head>\r\n");
out.write("<body>\r\n");
out.write("<h1>Index Page 입니다.</h1>\r\n");
out.write("<hr>\r\n");
out.write("<!-- JSP에 포함되면서 HTML태그가 적용된다. -->\r\n");
out.write("<footer>\r\n");
out.write("\tCopyright KH Corp. All rights reserved.\r\n");
out.write("</footer>");
out.write("\r\n");
out.write("</body>\r\n");
out.write("</html>");
```

## include 지시자 태그 실습

- 실습 테스트

index\_jsp.java

```
out.write("<!DOCTYPE html>\r\n");
out.write("<html>\r\n");
out.write("<head>\r\n");
out.write("<meta charset=\"UTF-8\">\r\n");
out.write("<title>Java Server Page 실습</title>\r\n");
out.write("</head>\r\n");
out.write("<body>\r\n");
out.write("<h1>Index Page 입니다.</h1>\r\n");
out.write("<hr>\r\n");
out.write("<!-- JSP에 포함되면서 HTML태그가 적용된다. -->\r\n");
out.write("<footer>\r\n");
out.write("\tCopyright KH Corp. All rights reserved.\r\n");
out.write("</footer>");
out.write("\r\n");
out.write("</body>\r\n");
out.write("</html>");
```

<%@ include file="footer.html" %>



# JSP Exception 처리

# JSP Exception 처리

- JSP 페이지에서 발생하는 Exception을 처리하기 위해서는 별도의 예외 처리 페이지를 지정한다.
- 하나의 JSP 페이지에 대한 예외 처리 페이지는 하나만 지정할 수 있기 때문에 예외마다 다른 예외 처리는 불가능하다.

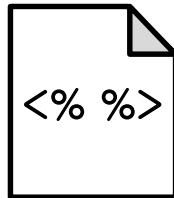
예외가 발생할 페이지

```
<%@page errorPage="/error/exceptionPage.jsp"%>
```

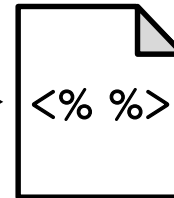
예외를 처리할 페이지

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

/throws\_error.jsp



/error/exceptionPage.jsp



----->

Web Container는 발생한 오류를 catch하여  
에러 처리 페이지로 전달한다.

# JSP Exception 처리

- 에러 발생 페이지 작성

error/throws\_error.jsp

```
<%@ page contentType="text/html; charset=UTF8"
    pageEncoding="UTF-8" errorPage="exceptionPage.jsp" %>
```

```
<html>
```

```
<head>
```

```
    <title>에러 발생 페이지!</title>
```

```
</head>
```

```
<body>
```

```
    <h2>JSP 에러 테스트 페이지</h2>
```

```
    <% String str = null; %>
```

```
    str 문자열 길이 : <%= str.length() %> 개
```

```
</body>
```

```
</html>
```

# JSP Exception 처리

- 에러 처리 페이지 작성

error/exceptionPage.jsp

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

```
<html>
```

```
<head>
```

```
  <title>Exception Check</title>
```

```
</head>
```

```
<body>
```

```
  <h2>에러가 발생하였습니다!</h2>
```

```
  <h3>에러 종류 : <%= exception.getClass().getName() %></h3>
```

```
</body>
```

```
</html>
```

# JSP Exception 처리

http://localhost:8800/first/views/error/throws\_error.jsp 접속



## JSP 에러 테스트 페이지

str 문자열 길이 : 4

String str = "TEST";



**에러가 발생하였습니다!**

에러 종류 : java.lang.NullPointerException

String str = null;

에러 발생 페이지 = web/views/error/throws\_error.jsp

에러 처리 페이지 = web/views/error/exceptionPage.jsp

# JSP 디버깅

## Translation Time Error

- JSP 파일을 Java 파일로 변환할 때 발생하는 오류
- Web Container가 JSP 페이지의 Scripting elements의 구문을 분석할 수 없을 때 발생한다  
예) ,%>' 없이 ,<%!' 만 사용한 경우

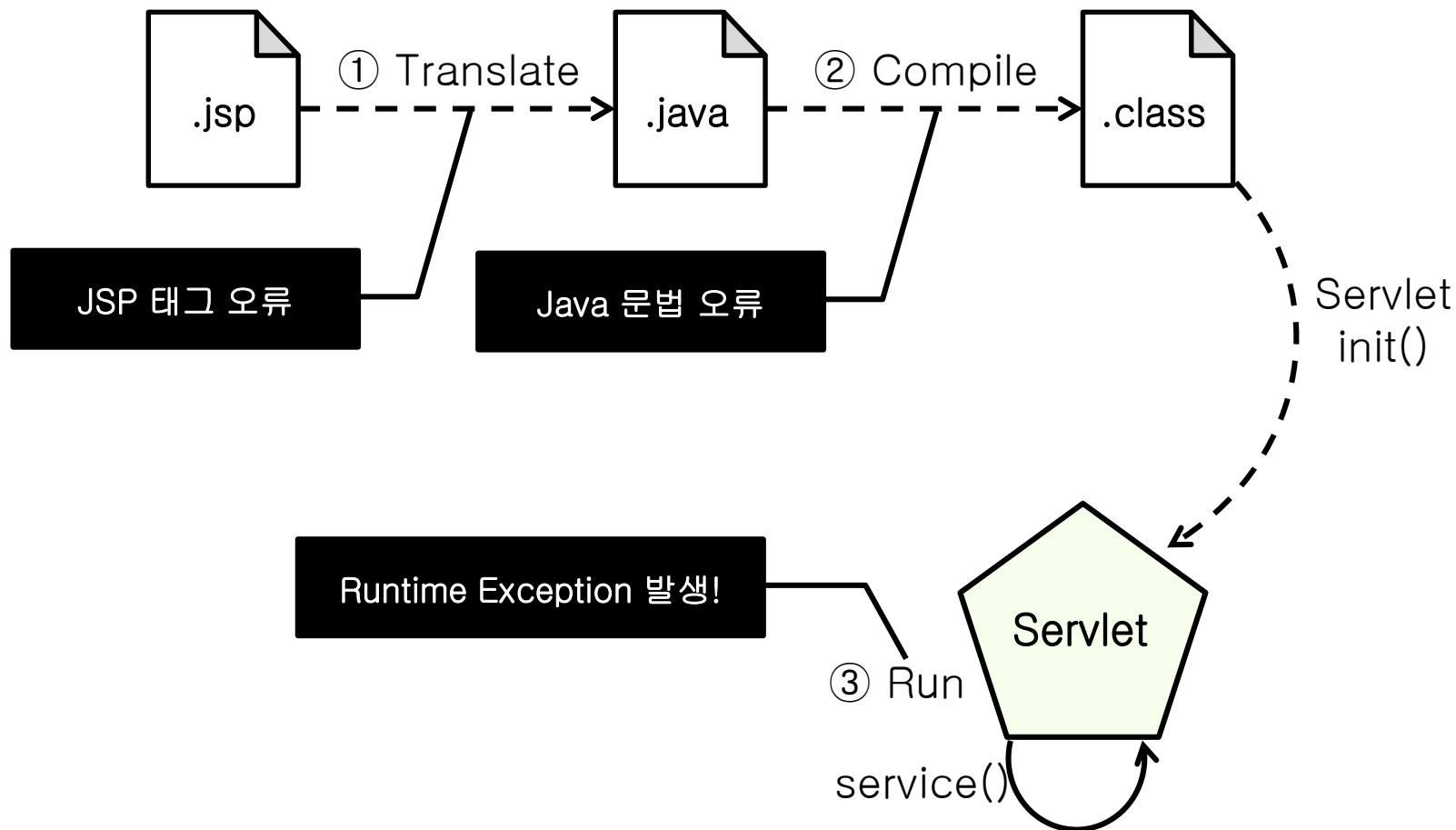
## Compile Time Error

- Java 파일을 Class 파일로 컴파일 할 때 발생하는 오류
- 자바 문법 오류  
예) 자바 문장을 끝낼 때, ';' 사용하지 않음

## Runtime error

- 실행 도중, Exception이 발생하는 경우  
예) HTML form 태그 전송 시 없는 이름을 getParameter() 인자에 사용하면, getParameter() 는 null을 return하는데 이 값을 가지고 조작하는 경우 NullPointerException 발생

## 에러 발생 시점





## Translation Time Error 실습

error/Translation\_error.jsp

```
<%@ page language="java" contentType="text/html;  
charset=UTF-8" pageEncoding="UTF-8"%>  
<html>  
  
<head>  
  <title>Translation Error</title>  
</head>  
  
<body>  
  <!-- 정상적으로 JSP 태그가 끝나지 않아 에러가 발생한다 --%>  
  <%= request.getSession()  
</body>  
</html>
```

## Translation Time Error 실습

← → ↺ ⓘ localhost:8800/first/views/error/Translation\_error.jsp

**HTTP Status 500 - /views/error/Translation\_error.jsp (line: 10, column: 4)**  
**Unterminated &lt;%= tag**

**type** Exception report

**message** /views/error/Translation\_error.jsp (line: 10, column: 4) Unterminated &lt;%= tag

**description** The server encountered an internal error that prevented it from fulfilling this request.

**exception**

org.apache.jasper.JasperException: /views/error/Translation\_error.jsp (line: 10, column: 4) Unterminated &lt;%= tag

```
org.apache.jasper.compiler.DefaultErrorHandler.jspError(DefaultErrorHandler.java:41)  
org.apache.jasper.compiler.ErrorDispatcher.dispatch(ErrorDispatcher.java:275)  
org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:91)  
org.apache.jasper.compiler.Parser.parseExpression(Parser.java:697)  
org.apache.jasper.compiler.Parser.parseElements(Parser.java:1443)  
org.apache.jasper.compiler.Parser.parse(Parser.java:139)  
org.apache.jasper.compiler.ParserController.doParse(ParserController.java:227)  
org.apache.jasper.compiler.ParserController.parse(ParserController.java:100)  
org.apache.jasper.compiler.Compiler.generateJava(Compiler.java:201)  
org.apache.jasper.compiler.Compiler.compile(Compiler.java:358)  
org.apache.jasper.compiler.Compiler.compile(Compiler.java:338)  
org.apache.jasper.compiler.Compiler.compile(Compiler.java:325)  
org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:586)  
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:363)  
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)  
test.common.filter.CharsetEncodingFilter.doFilter(CharsetEncodingFilter.java:33)
```

**note** The full stack trace of the root cause is available in the Apache Tomcat/8.0.45 logs.

## Compile Time Error 실습

error/Translation\_error.jsp

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

```
<html>  
<head>  
  <title>Compile Error</title>  
</head>
```

```
<body>  
<!-- 코드가 ';'으로 끝나지 않아 컴파일 에러가 발생한다 --%>  
<% out.print("Welcome :" ) %>  
</body>  
</html>
```

## Compile Time Error 실습

← → ↻ ⓘ localhost:8800/first/views/error/Compile\_error.jsp

### HTTP Status 500 - Unable to compile class for JSP:

**type** Exception report

**message** Unable to compile class for JSP:

**description** The server encountered an internal error that prevented it from fulfilling this request.

**exception**

```
org.apache.jasper.JasperException: Unable to compile class for JSP:

An error occurred at line: 11 in the jsp file: /views/error/Compile_error.jsp
Syntax error, insert ";" to complete Statement
8: </head>
9: <body>
10: <%
11:     out.print("Welcome :")
12: %>
13: </body>
14: </html>
```

Stacktrace:

```
org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:102)
org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:198)
org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:457)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:363)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:338)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:325)
org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:586)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:363)
```

## RunTime Error 실습

error/Runtime\_error.jsp

```
<%@ page language="java" contentType="text/html;  
charset=UTF-8" pageEncoding="UTF-8"%>  
  
<html>  
<head>  
<title>Runtime Error</title>  
</head>  
  
<body>  
<!-- 전달받지 않은 name 의 길이를 출력하므로 에러 발생 --%>  
<%= request.getParameter("name").length() %>  
</body>  
</html>
```

## RunTime Error 실습

← → ↻ ⓘ localhost:8800/first/views/error/Runtime\_error.jsp

---

**HTTP Status 500 - An exception occurred processing JSP page /views/error/Runtime\_error.jsp at line 10**

---

**type** Exception report

**message** An exception occurred processing JSP page /views/error/Runtime\_error.jsp at line 10

**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 /views/error/Runtime\_error.jsp at line 10

```
7: <title>Runtime Error</title>
8: </head>
9: <body>
10: <%= request.getParameter("name").length() %>
11: </body>
12: </html>
```

Stacktrace:

```
org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:579)
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)
test.common.filter.CharsetEncodingFilter.doFilter(CharsetEncodingFilter.java:33)
```

**root cause**

```
java.lang.NullPointerException
    org.apache.jsp.views.error.Runtime_005error_jsp._jspService(Runtime_005error_jsp.java:117)
    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)
    test.common.filter.CharsetEncodingFilter.doFilter(CharsetEncodingFilter.java:33)
```

**note** The full stack trace of the root cause is available in the Apache Tomcat/8.0.45 logs.