

# 1.JSP

목차

## 1. JSP

### 1.1 JSP란?

### 1.2 JSP Life Cycle

### 1.3 JSP Scripting Elements

#### 1.3.1 Comment

#### 1.3.2 Directive(page, include, taglib)

#### 1.3.3 Declaration

#### 1.3.4 Scriptlet

#### 1.3.5 Expression

### 1.4 JSP Implicity Object

#### 1.4.1 request object

#### 1.4.2 response object

#### 1.4.3 page object

#### 1.4.4 pageContext object

#### 1.4.5 application object

#### 1.4.6 config object

#### 1.4.7 session object

#### 1.4.8 out object

#### 1.4.9 exception

## 2. Action In JSP

### 2.1 <jsp:useBean>과 <jsp:getProperty>

### 2.2 <jsp:setProperty>

### 2.3 <jsp:include>

### 2.4 <jsp:forward>

## 1. JSP(Java Server Page)소개

- JSP는 MS의 asp(Active Server Page)에 대항하기 위해서 만들어진 자바 기술이다.  
MS에서 개발된 asp는 CGI를 대체하기 위해 개발되었는데, 프로그램 개발이 쉽고, 성능도 CGI보다 뛰어나다.  
이러한 장점들을 자바 진영에서는 JSP가 제공한다. **JSP는 기본적으로 서블릿 기술을 바탕으로 이루어져 있다.**  
JSP는 서블릿의 상위개념으로 볼수 있다. 서블릿에 비해 장점을 갖는 jsp에 대해 알아보자

- ① JSP는 서블릿 보다 쉽고, 빠르게 프로그램을 작성할 수 있다.
- ② JSP는 화면 구성을 위한 HTML부분과 프로그램 로직을 분리할 수 있기 때문에 프로그래머와 웹 디자이너가 서로 협력해서 작업하기 쉽다.(자바빈즈, jsp태그)
- ③ JSP엔진은 효율적인 코드를 생성하기 때문에 서블릿보다 성능면에서 뛰어나다.
- ④ 서블릿은 자바 프로그래밍이기 때문에 HTML의 페이지 디자인을 하기 위한 도구가 부족한데 비해, JSP는 HTML페이지 디자인 도구들을 이용해서 작성할 수 있다.

	CGI/Perl	Mod_Perl	ASP	JSP
웹서버	모든웹서버	아파치 웹서버	MS IIS, PWS	모든 웹서버
서버와 플랫폼이식성	X	X	X	O
재사용성 /코드의모듈	X	X	X	O
스크립트 언어	C, Perl	Perl	VB Script,Jscript	Java
메모리 유출방지	O	X	X	O
병행성 지원	X	O	O	O

[표 1 JSP와 다른 기술 비교]

- cf) 초당 서비스되는 시간과 요청이 처리되는 시간을 asp와 jsp를 비교하면  
약 2 배정도 차이가 난다.  
단 각각의 엔진에 따라 달라질 수 있음을 기억하기 바란다.

### - 간단한 jsp프로그램을 작성해 보자.

JSP 페이지는 html 태그만 사용할 수 있는 것이 아니라 자바 프로그램 코드를 삽입할 수 있다.

<% %>태그 사이에 자바코드를 기술하면 된다.

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

```
<HTML>
```

```
<HEAD><TITLE>Hello World</TITLE></HEAD>
```

```
<BODY><H2>Hello World : 헬로월드</H2>
```

```
오늘의 날짜와 시간은 : <%= new java.util.Date() %>
```

-주석처리 : html주석과 jsp에서의 주석의 차이는 일단 브라우저에서 소스보기를  
주석처리한 것이 보이는지 보이지 않는 지의 차이이다.  
Jsp에서의 주석처리는 컴파일시 주석 자체를 컴파일하지 않기 때문에  
보이지 않는다.

①Html 주석

<!-- *html에서의 주석처리* →

②JSP 주석

<%-- *jsp에서의 주석처리* --%>

comment.jsp

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

<HTML>
<HEAD><TITLE>주석 테스트 </TITLE></HEAD>
<BODY>
<!-- 이 주석은 클라이언트에서 볼 수 있습니다. -->

<H2>JSP 주석 테스트</H2>

<%-- 이 주석은 클라이언트에서는 볼 수 없습니다. --%>
```

## 2.JSP의 처리과정 및 라이프 사이클

- JSP 페이지는 jsp 컨테이너(엔진)에 의 서블릿 코드로 변환된다.  
즉 아래의 단계를 거쳐 실행된다.

1 서블릿코드 생성단계 : JSP 페이지가 서블릿 프로그램 코드로 변환된다.

이 단계는 jsp페이지가 처음으로 클라이언트의 요청을 받을 때 수행되고,한번 서블릿코드가 생성되면,jsp페이지 내용이 변경되지 않는 한 이 단계는 수행되지 않는다.( **JSP변경시에만 수행**)

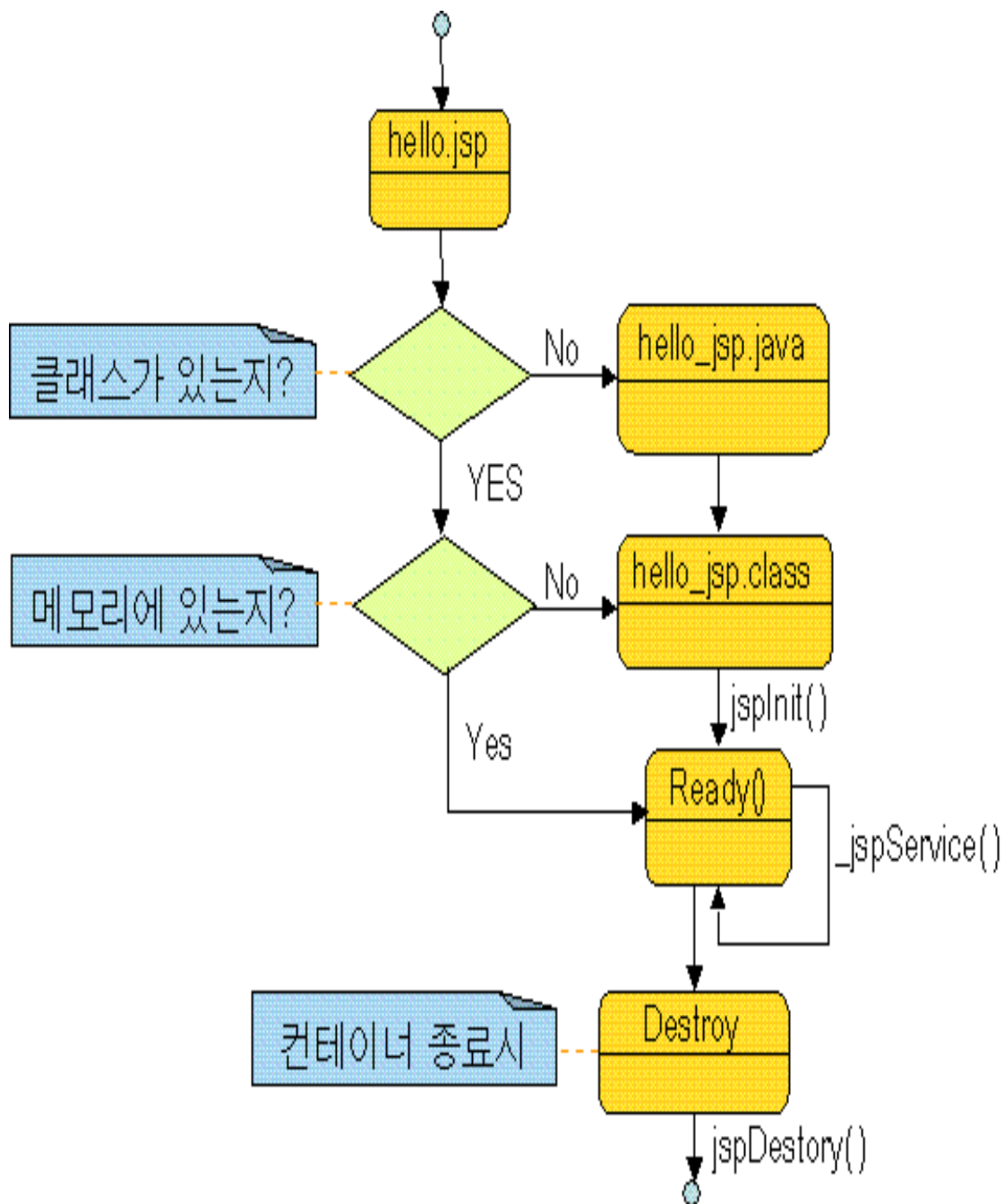
2 컴파일단계 : 서블릿 코드 생성 단계에서 생성된 서블릿 코드는 컴파일 단계를 거쳐 컴파일된다. 컴파일할때 jdk의 tools.jar에 포함되어 있는 자바 컴파일러가 사용된다.

3 요청 처리단계 : 요청 처리 단계는 클라이언트의 요청을 받아서 처리하는 단계이다.  
요청 처리는 컴파일 단계에서 생성된 서블릿클래스를 통해이루어진다.

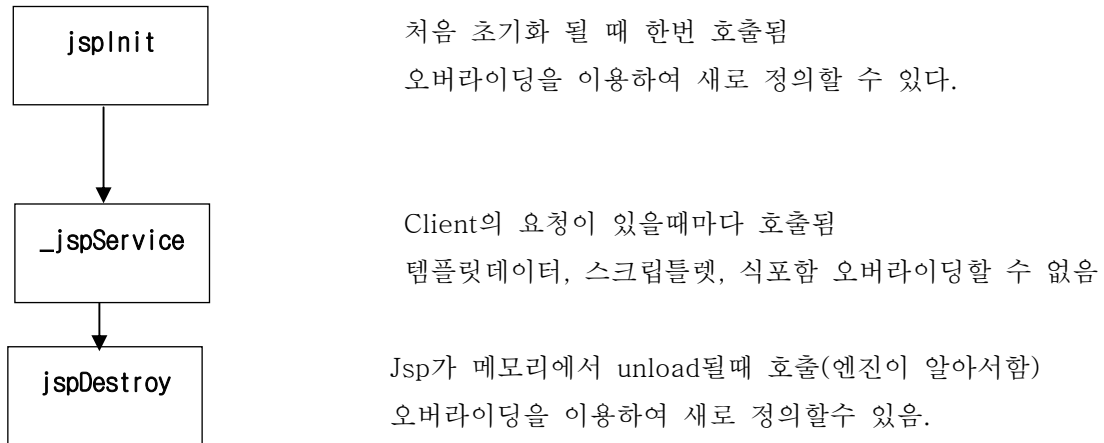
hello.jsp

```
<%@ page contentType="text/html;charset=euc-kr" %>
<HTML>
    <HEAD><TITLE>라이프사이클</TITLE></HEAD>
    <BODY>
        <H2>HELLO JSP !!!!! </H2>
    </BODY>
</HTML>
```

## hello.jsp 처리과정



- JSP(객체) 라이프사이클 : JSP의 라이프사이클은 서블릿에서와 마찬가지로 유사하다.



### 3. JSP 페이지의 구성요소

JSP 구성 분류	종류
지시어 (Directives)	<code>&lt;%@ page %&gt;</code> <code>&lt;%@ include %&gt;</code> <code>&lt;%@ taglib %&gt;</code>
스크립팅요소 (Scripting Elements)	<code>&lt;% %&gt;</code> (스크립틀릿) <code>&lt;%! %&gt;</code> (선언) <code>&lt;%= %&gt;</code> (익스프레션) 식
표준 액션 태그 (Standard Actions)	<code>&lt;jsp:useBean&gt;</code> <code>&lt;jsp:getProperty&gt;</code> <code>&lt;jsp:setProperty&gt;</code> <code>&lt;jsp:param&gt;</code> <code>&lt;jsp:include&gt;</code> <code>&lt;jsp:forward&gt;</code> <code>&lt;jsp:plugin&gt;</code>
주석(Hidden Comments)	<code>&lt;%-- --%&gt;</code>

3-1 .스크립팅 요소(Scripting Elements): 스크립팅 요소 3 가지가 있습니다.

선언(!), 스크립트릿, 식(=)입니다.

선언은 메소드나 변수의 선언을 하고,

스크립트릿은 jsp문장 들의 모음을 위한 것이라고 생각하시고,

식은 자바언어의 식과 같다.

①스크립트릿(scriptlet) : jsp페이지에서 자바 문장을 기술하는 부분입니다.

형태 : <% \_jspService() 메소드안에 들어가는 자바코드기술 %>

ex> first.jsp

②선언(declarelation) : 멤버변수나 메소드를 선언하기 위해서 사용된다. Jsp페이지에서 초기화될 때

다른선언문장이나, 스크립트릿,식에서 사용할 수 있다.

형태: <%! 선언할 멤버변수나 멤버메소드를 적습니다. %>

ex> lifecycleCounter.jsp

③표현식(expression) : 식은 계산식이나 변수, 메소드호출등을 기술할 수 있는 것이다.

계산식과 함수의 결과는 String형으로 변환되어 out(JspWriter)변수를 통해

클라이언트에 전달된다. 만약 String으로 변환이 되지 않으면

ClassCastException이 발생한다. 주의할 점은 세미콜론(;)을 사용하지

않는다는 것이다.

형태: <%= 메소드호출, 변수 %>

3-2. 디렉티브(Directives,지시어) : 지시어에는 3 가지가 있습니다.

① page 지시어 : jsp 페이지에 관련된 속성을 정의하고 이 속성들을 jsp

엔진에 정보를 제공한다. Page지시어는 페이지당 1 번이상

올 수 없지만 import는 2 번 이상 올 수있다.

또한 페이지의 어느곳에 위치해도 상관없지만 일반적으로 가장위에 놓는다.

형태: <%@ page 속성 1="속성값 1" 속성 2="속성값 2"... %>

예)

<%@page language="java"

```
import="java.io.*"
buffer="16kb"
errorPage="/error.jsp"
info="test"
contentType="text/html; charset=EUC-KR"
```

%>

위의 뜻은 페이지에서 사용할 언어는 자바이고  
java.io패키지를 импорт하고,  
출력을 위한 버퍼는 16k이며  
페이지에서 에러가 발생하면 error.jsp로 제어가 이동하고,  
간략한 페이지정보는 test 이다 라는의미이다.

### Page지시어 주요속성

속 성	설 명	기본값
language	스크립트 언어를 지정한다.	java
Import	jsp 파일 내에서 사용할 외부 자바 패키지나 클래스 지정	
session	세션 생성 여부 지정	true
Buffer	버퍼 크기 지정	8kb
autoFlush	버퍼 내용 자동 비움 지정	true
isThreadSafe	단일 스레드 모델을 사용하여 동시성 제어 여부 지정	true
Info	JSP 페이지 설명	
errorPage	에러 발생 시 호출 페이지 지정	
isErrorPage	에러만 처리하는 페이지 지정	false
contentType	MIME 형식과 캐릭터셋 설정	text/html; charset=ISO-8859-1

ex> error\_test.jsp,error.jsp

②include 지시어:외부 데이터(html,jsp,text)를 jsp페이지에 포함 시킬 때 사용하는 기법이다.

외부 데이터(html,jsp)를 jsp페이지에 포함 시킬 때는  
include지시어나 jsp action태그<jsp:include>를 사용하는데  
차이점은 include지시어의 경우는 file code를포함하여  
컴파일하고, 포함하는 파일 변경시 다시 컴파일하는 반면  
jsp action태그는 독립적으로 컴파일된후 클라이언트의



요청이 있을 때 외부데이터를 포함한다.  
따라서 include지시어는 정적인 데이터포함,  
jsp태그는 동적인 데이터를 포함하는 용도로 사용하면된다.

형태: <%@include file="상대URL의 파일명" %>

4 **tablib지시어** : jsp에서는 사용자가 원하는 태그를 새로 생성해서 사용할 수  
있습니다. 이러한 기능을 제공하는 것이 태그 라이브러리입니다.  
다른 말로는 커스텀태그 라고도합니다.

#### ex1>JSP에서 form데이터 처리

- JSP에서의 Form데이터 처리는 서블릿에서 하는 방법과 거의 동일하다.  
간단한 예제를 통해 알아보자.

Ex>form1.html,join.jsp

#### ex2>응용프로그램 데이터 보내기

- jsp에서도 응용프로그램에 데이터를 보내는 방법은 서블릿과 동일합니다.  
데이터의 MIME타입을 Page지시어를 이용해서 지정할 수 있습니다.  
또한 jsp에서는 page지시어 이외에 response객체에서 setContentType()메소드를  
이용해서 MIME 타입을 지정할수 있습니다.

📁 Page지시어를 이용하는 경우

ex>salary.jsp

```

<%@ page contentType="application/vnd.ms-excel" %>
<%@ page contentType=" application/msword " %>
<%--
response.setContentType("application/vnd.ms-excel");
response.setContentType("application/msword");
--%>
<html>
<body>
<table>
<table border=1 width=90%>
<
    <tr><th>이름</th><th>월급</th><th>주소</th></tr>
    <tr><td>교육원</td><td>500</td><td>서울시 영등포구</td></tr>
    <tr><td>김경호</td><td>350</td><td>서울시 영등포구</td></tr>
    <tr><td>여러분 </td><td>350</td><td>서울시 영등포구</td></tr>
</table>
</body>
</html>

```