

JSP

2강. JSP태그

- jsp태그 종류 4가지
- jsp아키텍처

Servlet

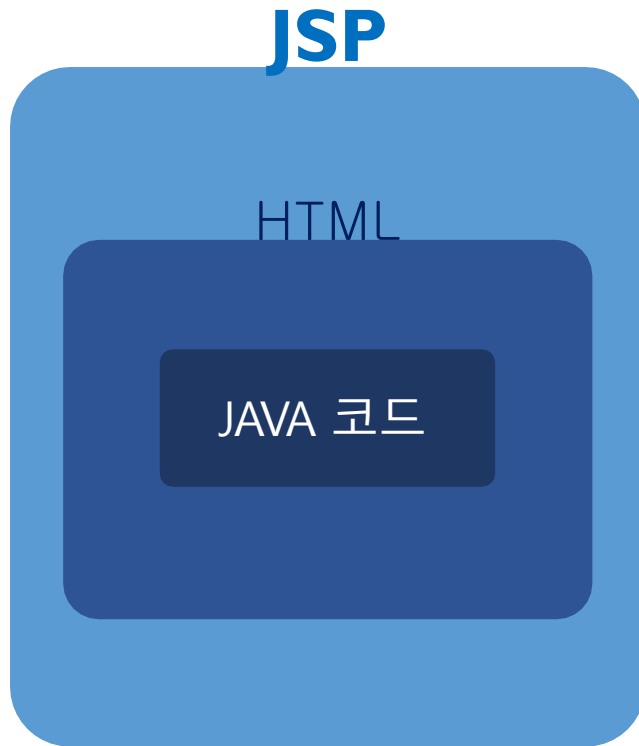
JAVA 코드

VS

JSP

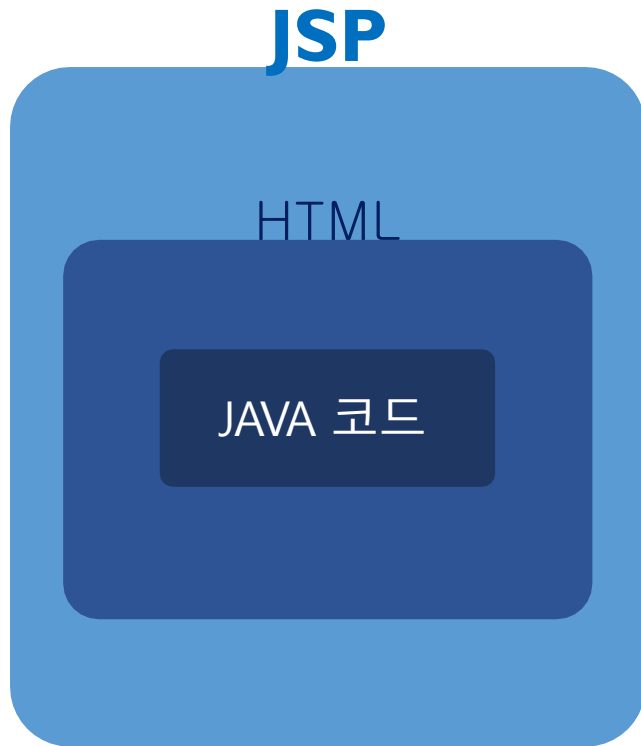
HTML

JAVA 코드



- JSP 특징

1. 동적 웹어플리케이션 컴포넌트 (자바 코드 사용가능)
2. jsp 확장자
3. 클라이언트의 요청에 동적으로 작동하고, 응답은 HTML을 이용.
4. JSP는 웹서버와 통신시에 자동으로 Servlet으로 변환됨.
5. MVC패턴에서 View로 이용됨.



* JSP 태그의 개념 이해

- Servlet은 java언어를 이용하여 문서를 작성하고, 출력객체 (PrintWriter)를 이용하여 html 코드를 삽입합니다.
- jsp는 서블릿과 반대로 html코드에 java언어를 삽입하여 동적 문서를 만들 수 있습니다.
- html코드 안에 java코드를 삽입하기 위해서는 **jsp태그**를 이용해야 하며, 이러한 **태그**를 학습해야 합니다.

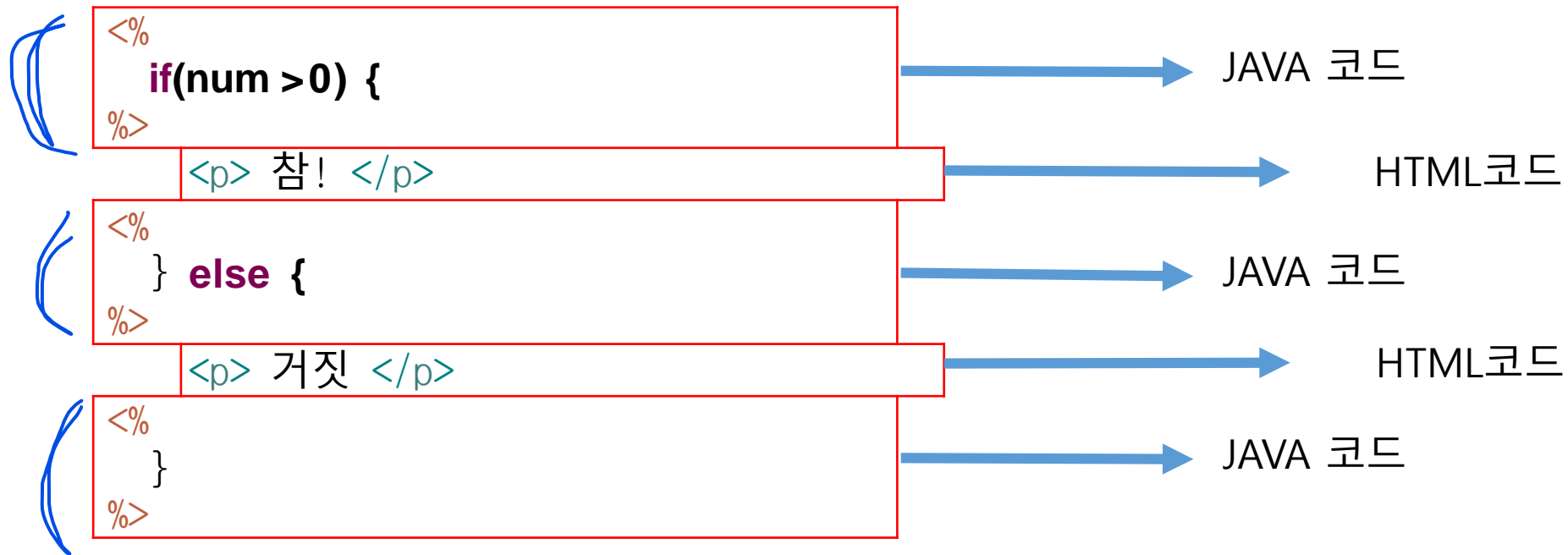
JSP 태그의 종류

JSP태그 종류	생김새	특징
스크립트릿(scriptlet)	<code><% %></code> ☆	모든 자바 코드 기술 가능.
지시자(directive)	<code><%@ %></code> ☆	페이지 속성을 지정.
선언자(declaration)	<code><%! %></code>	변수나 메서드 선언 시 사용.
표현식(expression)	<code><%= %></code> ☆	결과 값을 출력할 때 사용.
주석(comments)	<code><%-- --%></code>	코드 주석처리 시에 사용.

* 스크립트릿(Scriptlet)

- JSP페이지에서 JAVA언어를 사용하기 위한 요소 중 가장 많이 사용하는 태그입니다.
- 스크립트릿 안에는 우리가 알고 있는 거의 모든 JAVA 코드를 사용할 수 있습니다.

스크립트릿



* 선언(Declaration)

- JSP 페이지 내에서 사용되는 변수 또는 메서드를 선언할 때 사용합니다.
- 여기 선언된 변수 및 메서드는 전역의 의미(멤버변수)로 사용됩니다.



선언

<%!

```
int num=10;
```

```
String str = "변수선언";
```

```
ArrayList<String> list = new ArrayList<String>();
```

```
public void method() {
```

```
    System.out.println("메서드 선언");
```

```
}
```

%>

변수 선언 초기화

객체 생성 초기화

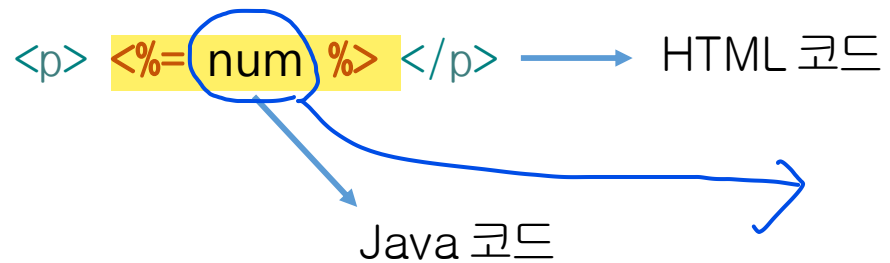
메서드 선언

* 표현식(Expression)

- JSP 페이지 내에서 사용되는 변수의 값 또는 메서드 호출 결과 값을 출력하기 위해 사용됩니다.
- 결과 값의 데이터 유형은 String이며, 세미콜론(;)을 사용할 수 없습니다.
- 표현식은 `out.println()`을 대체합니다.

```
<%  
    int num = 10;  
    String str = "안녕하세요";  
%>
```

표현식



* 주석(Comments)

- 실제 프로그램 실행에는 영향이 없고, 프로그램 설명 등의 목적으로 사용되는 태그입니다.
- HTML 및 JSP 주석이 각각 별도로 존재합니다.
- HTML 주석은 `<!-- -->` 표기하고, 웹 브라우저에서 페이지 소스보기를 하면 주석도 표기됩니다.
- JSP 주석은 `<%-- --%>` 표기하고, 웹 브라우저에서 소스보기를 해도 나타나지 않습니다.
- JAVA의 주석도 혼용 사용 가능합니다.(// , /* ~~~ */)

주석

`<!-- 주석 태그 -->`

HTML 주석

`<%-- Hello JSP World!! --%>`

JSP 주석

< 1. @

* 지시자(directive)

- JSP페이지의 전체적인 속성을 지정할 때 사용합니다.

1. **page**: JSP페이지에 대한 정보를 지정한다. JSP가 생성하는 문서의 타입, импорт, 에러 페이지 등 페이지에서 필요로 하는 정보를 설정합니다.

- **페이지 디렉티브에 선언 될 수 있는 속성들**

a. **language** - 언어를 지정합니다. java만 지정할 수 있습니다.

b. **import** - 패키지를 import할 때 사용합니다.

c. **errorPage** - 설정 된 페이지에서 에러가 발생했을 시에 미리 만들어 둔 에러 페이지를 호출합니다.

d. **contentType** - text/html이라고 쓰면 결과가 HTML문서로 만들어집니다.

e. **pageEncoding** - 출력 문자 인코딩 입니다. 한글을 사용하려면 utf-8로 지정합니다.

2. **include**: JSP 페이지의 특정 영역에 다른 문서를 포함시킵니다.

3. **taglib**: JSP 페이지에서 사용할 태그 라이브러리를 지정합니다.

지시자

예시)

1) page : 페이지 기본 설정 → <% page 속성="속성 값">

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

2) include : include file 설정 → <% include file="파일명">

```
<%@ include file="header.jsp"%>
```

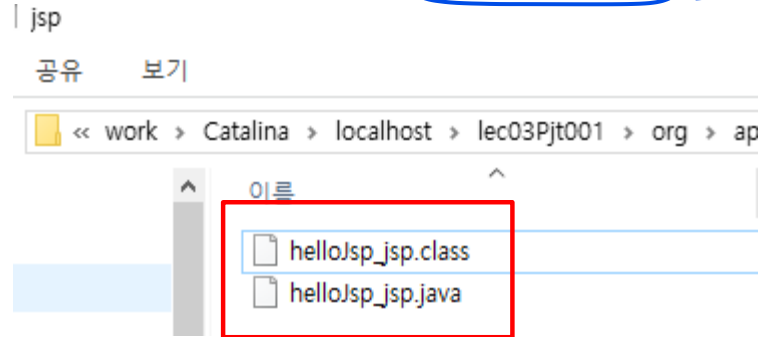
3) taglib : 외부 라이브러리 태그 설정 → <% taglib uri="uri" prefix="네임스페이스명">

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

- JSP 아키텍처

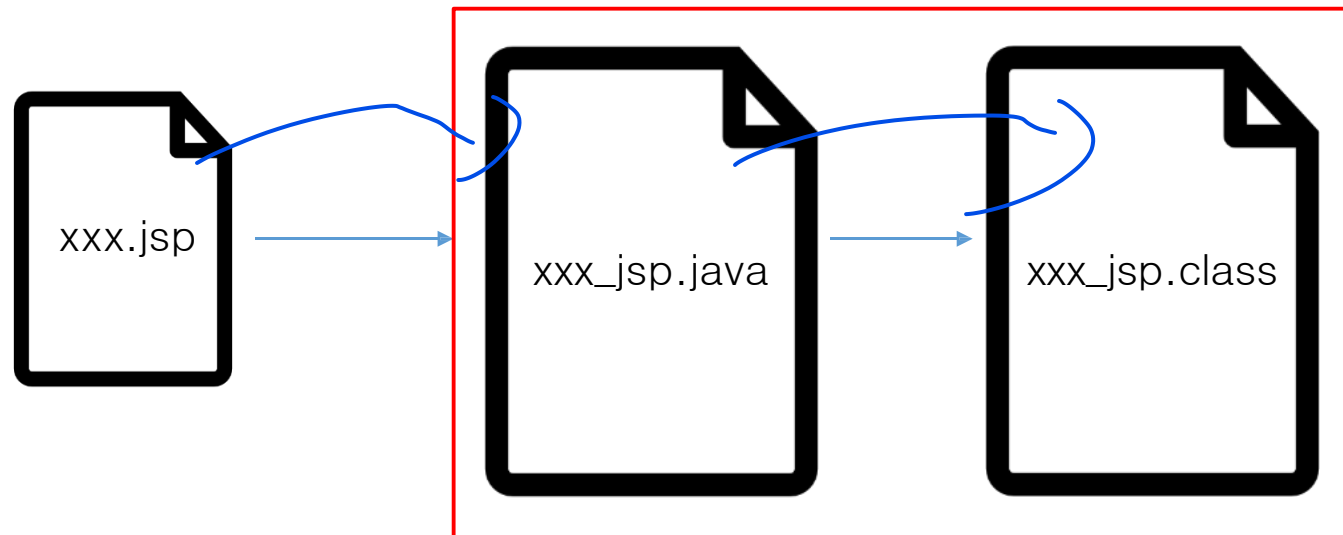
was

: .jsp파일을 실행(요청)하면 웹 서버에서 우선 jsp파일을 java코드로 변환합니다. 그 이후 그 파일을 컴파일하고 html로 응답합니다.

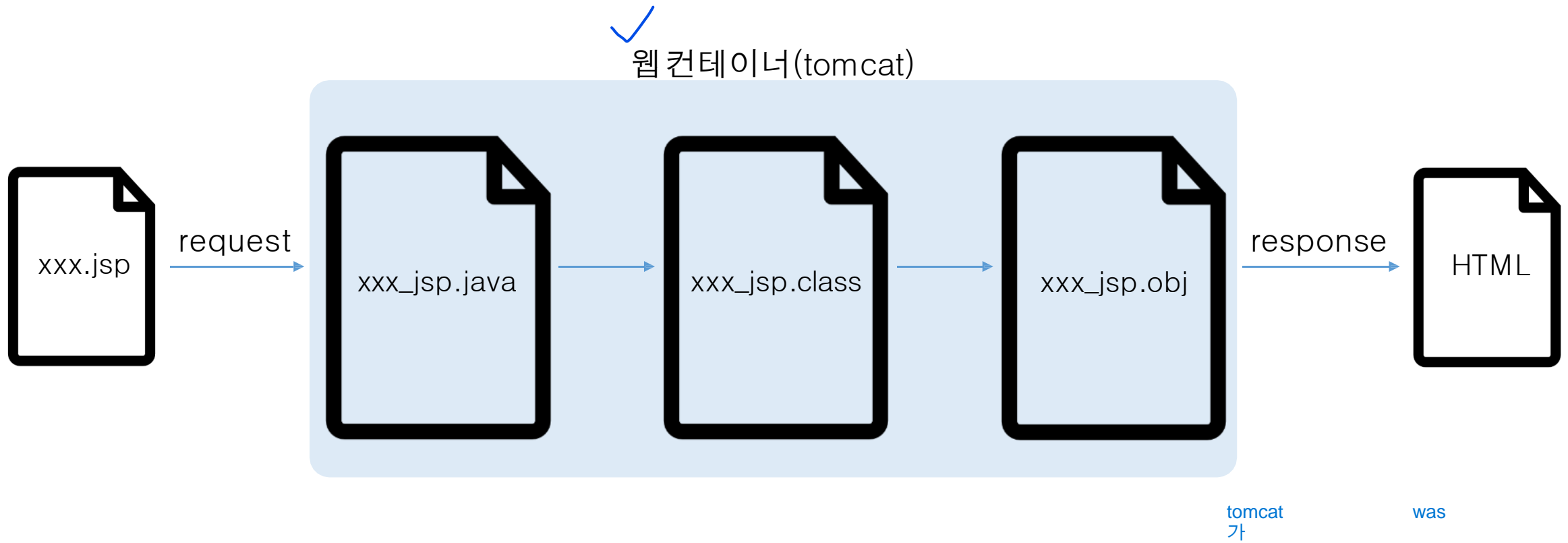


was(8181)

C:\Wjsp\download\Wapache-tomcat-9.0.24\work\Catalina\localhost\WJSPObject\Worg\Wapache\Wjsp



✓ - JSP 아키텍처

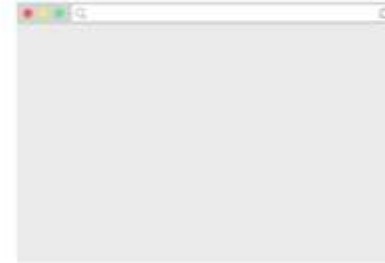


- 그래서 서버사이드 렌더링이 뭔데!!!!??

내가 다 그려서 줄게!
JSP

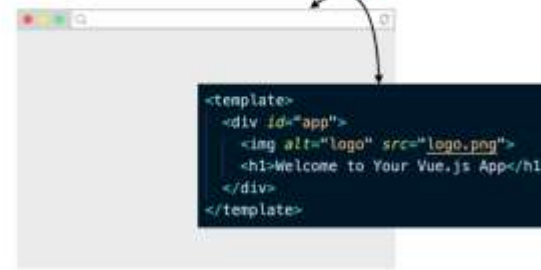
```
<template>
<div id="app">
  
  <h1>Welcome to Your Vue.js App</h1>
</div>
</template>
```

서버 사이드 렌더링(SSR)



클라이언트 사이드 렌더링(CSR)

data



데이터만 보내
화면은 내가 그린다!
React
Vue
angular